Extracting and Analyzing Key Information From Scanned Prescriptions using ML

A PROJECT REPORT

Submitted by

C RUSHI KESAVA NAIDU	**-20201CSD0138**
G YASWANTH	**-20201CSD0133**
E VAMSI KAMAL	**-20201CSD0142**
N VINAY	**-20201CSD0156**

Under the guidance of,

Mrs. SHAIK SALMA BEGUM
(Assistant professor,B.tech, M.tech,PhD)

in partial fulfillment for the award of the

degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING,
(DATA SCIENCE)

At

GAIN MORE KNOWLEDGE
REACH GREATER HEIGHTS

PRESIDENCY UNIVERSITY

BENGALURU

JANUARY 2024

PRESIDENCY UNIVERSITY

SCHOOL OF COMPUTER SCIENCE & ENGINEERING &DATA SCIENCE

CERTIFICATE

This is to certify that the Project report **"Extracting and Analyzing Key Information From Scanned Prescriptions using ML"** being submitted by "C RUSHI KESAVA NAIDU" bearing roll number "20201CSD0138", "G YASWANTH" bearing roll number "20201CSD0133","E VAMSI KAMAL" bearing roll number "20201CSD0142", "N VINAY" bearing roll number "20201CSD0156" , in partial fulfilment of requirement for the award of degree of Bachelor of Technology in Computer Science and Engineering is a bonafide work carried out under my supervision.

Mrs. SHAIK SALMA BEGUM
Assistant Professor,
Project Guide
School of CSE&IS
Presidency University

Dr. A JAYACHANDRAN
Associate Professor & HoD
School of CSE & IS
Presidency University

Dr. C. KALAIARASAN
Associate Dean
School of CSE&IS
Presidency University

Dr. L. SHAKKEERA
Associate Dean
School of CSE&IS
Presidency University

Dr. SAMEERUDDIN KHAN
Dean
School of CSE&IS
Presidency University

PRESIDENCY UNIVERSITY

SCHOOL OF COMPUTER SCIENCE ENGINEERING & DATA SCIENCE

DECLARATION

We hereby declare that the work, which is being presented in the project report entitled **Extracting and Analyzing Key Information From Scanned Prescriptions using ML** in partial fulfilment for the award of Degree of **Bachelor of Technology** in **Computer Science and Engineering**, is a record of our own investigations carried under the guidance of Mrs.Shaik Salma Begum**,**Assistant Professor **School of Computer Science Engineering & Information Science, Presidency University, Bengaluru.**

We have not submitted the matter presented in this report anywhere for the award of any other Degree.

Students Names	**Roll Number**	**Signature's**
C Rushi Kesava Nandu	20201CSD0156	
N Vinay	20201CSD0156	
E Vamsi Kamal	20201CSD0142	
G Yaswanth	20201CSD0133	

ABSTRACT

The project aims to improve healthcare by utilizing Machine Learning (ML) techniques for prescription analysis. It aims to streamline the data extraction process, enhance prescription digitization, and contribute to a more efficient healthcare ecosystem. The project addresses challenges in traditional manual transcription of prescriptions, such as varied handwriting styles and medical abbreviations. By using Natural Language Processing (NLP) and Optical Character Recognition (OCR), the project aims to automate the extraction of critical information such as patient details, prescribed medications, dosage instructions, and prescribing physician details. The methodology involves preprocessing scanned prescriptions to improve image quality and facilitate accurate OCR. The NLP component focuses on understanding medical terms, recognizing patterns in dosage instructions, and associating medication information with the patient. The project utilizes a robust dataset to train and fine-tune ML models. The project's significance lies in its potential to revolutionize prescription management systems, reducing transcription errors and improving data accuracy, ultimately contributing to improved patient safety and more effective healthcare delivery.

"Extracting and Analyzing Key Information from Scanned Prescriptions using ML" project signifies a significant step towards a technologically advanced and data-driven healthcare landscape. By combining OCR and NLP technologies, we aim to alleviate the burden of manual prescription transcription, contributing to improved patient care, enhanced prescription data analysis, and ultimately, a more efficient healthcare system. As technology continues to evolve, this project establishes a foundation for further innovations in medical data management, with far-reaching implications for the future of healthcare.

ACKNOWLEDGEMENT

First of all, we indebted to the **GOD ALMIGHTY** for giving me an opportunity to excel in our efforts to complete this project on time.We express our sincere thanks to our respected dean **Dr. Md. Sameeruddin Khan**, Dean, School of Computer Science Engineering & Information Science, Presidency University for getting us permission to undergo the project.We record our heartfelt gratitude to our beloved Associate Deans **Dr. Kalaiarasan C and Dr. Shakkeera L,** School of Computer Science Engineering & Information Science, Presidency University and **Dr. A Jayachandran,** Head of the Department, School of Computer Science Engineering & Information Science, Presidency University for rendering timely help for the successful completion of this project.

We are greatly indebted to our guide **Mrs. Shaik Salma Begum**, **Assistant Professor**, School of Computer Science Engineering & Information Science, Presidency University for her inspirational guidance, and valuable suggestions and for providing us a chance to express our technical capabilities in every respect for the completion of the project work.

We would like to convey our gratitude and heartfelt thanks to the University Project-II Coordinators **Dr. Sanjeev P Kaulgud, Dr. Mrutyunjaya MS** and also the department Project Coordinators **Dr.Manjula H M**.

We thank our family and friends for the strong support and inspiration they have provided us in bringing out this project.

C Rushi Kesava Nandu

N Vinay

E Vamsi Kamal

G Yaswanth

LIST OF TABLES

LIST OF FIGURES

TABLE OF CONTENTS

CHAPTER-1
INTRODUCTION

This Software Requirements Specification provides a complete description of all the functions and specifications of the OCR for Printed and Hand written documents. This documentation presents an intense study of requirements of OCR system where we can scan hardcopy of documents in scanner and store its data in database after processing thus reducing the manual burden of entering the data in database.

Since the beginning of optical character recognition, numerous institutions have started digitising their historical archives in an effort to provide accessibility to historical materials. The old, digitised papers may include deteriorations from outdated printing techniques or from time-related wear and tear. Frequently seen visual characteristics on the documents include style and font differences, broken characters, noise levels, ink intensity, and damage from tearing and folding, among other things. Many of these characteristics can result in unsuccessful character recognition since they are unfavourable to contemporary optical character recognition systems. In order to enhance character recognition performance, this study employs image processing techniques to address the stated challenge. Additionally, typical picture quality traits of historical documents that have had their text digitised and rendered unidentifiable are examined. The Tesseract programme, which is available for free, was utilised as the optical character recognition tool for this study. The historical records for Tesseract were prepared using image processing techniques such as Gaussian lowpass filtering, Otsu's optimal thresholding method, and morphological processes. When the OCR output was assessed using the Precision and Recall classification method, it was seen that the precision had increased by 18 percentage points and the recall had improved by 63 percentage points. This demonstrates the effectiveness of employing picture pre-processing techniques to improve historical document readability for OCR software. Additionally, it was observed that prevalent traits that are particularly detrimental to Tesseract include font irregularities, the occurrence of character fading, broken characters, and Poisson noise.

1.1 Aim of the project :

The purpose of this work is to determine potential causes for OCR algorithm failures when applied to old historical documents. An analysis and presentation are made of the image similarities that did not lead to successful recognition. It will also be investigated whether different image processing methods used to produce photos with typical visual flaws yield better OCR results. There will be a comparison of the OCR results of unprocessed and processed images.

1.2 Project Scope :

Almost any kind of physical document may now be scanned and treated like a digital one, however a raw scan of a document typically lacks metadata. At first, it's just a high-quality image of a real page, but with the help of software tools, it can be converted into a file where each figure, paragraph, and table becomes a separate composite that together make up the entire document. The handling of historical records will be greatly impacted by this. A vast amount of old historical documents have been scanned thanks to scanning technology, which allows any physical document to be digitalized. Old Australian and Finnish newspapers, British historical medical texts, and much more are included. A strategy to increase the accessibility of these documents would be to use an Optical Character Recognition (OCR) tool to extract the textual data that they contain.

Characters on an image can be converted to digital characters through a method called optical character recognition. Numerous jobs can be made simpler and more content can be accessed and searched by using OCR. The tool functions optimally when the image to be read satisfies specific quality requirements, such as high image quality, well-lit areas, readable text portions, and more. Errors still happen even when those prerequisites are satisfied. Thus, those requirements are frequently not met when an old document has aesthetic flaws.

Since historical archived form-type documents have unique form values, the problem is considerably more severe. These variables, which may be a name, a unique identification number, or anything else, were probably not automated in ancient form-type papers, which is why they varied graphically. Characters that are doubled, overlapped, or appear in lines are frequently observed in instances where typewriters or similar devices were in use. OCR is a fantastic tool, but it is not perfect.

1.3 Project Basic Requirements

1.3.1 Hardware Requirements

- Hardware :Processor Intel dual core and above
- Clock speed :3.0 GHz
- RAM size :512 MB
- Hard Disk capacity :400 GB
- Monitor type :15 inch color monitor

1.3.2 Software Requirements

- Operating System :Windows XP, Windows 7, Windows 8,Windows 10
- Application :HTML, CSS, JS, JSP, SERVLET
- Browser :Google chrome, Firefox
- Database :Google Firestore.
- Documentation :MS-Office

Layout analysis is often a performance-limiting step of optical character recognition (OCR) systems since the errors made at this stage propagate to all further stages of the system. Typical use cases of OCR are desktop scanning and large volume document conversion. Although, the layout analysis components of existing commercial and research OCR systems are powerful enough to meet the expectations of many users, they do not fulfill the requirements of large scale digitization tasks. This dissertation presents a high performance layout analysis system that satisfies the demands of both application scenarios. The layout analysis system has a robust and highly accurate generic layout analysis component that addresses the needs of desktop scanning applications, and a statistically motivated, style- directed, and trainable layout analysis component that is specifically designed to fulfill the requirements of large-scale document analysis applications.

Paper documents like books, handwritten manuscripts, magazines, newspapers, etc. have traditionally been used as the main source for acquiring, disseminating, and preserving knowledge. The advent of personal computers has given birth to another class of documents called electronic documents. An electronic document is a representation of a document using data structures that can be understood by computers. Typical examples of electronic documents are PDF, Word, XML, E-mails, Web pages, etc.

• Noise removal is a process that tries to detect and remove noise pixels in a document that are introduced by scanning or binarization.

• Skew correction is a process that detects and corrects the deviation of a document's orientation angle from the horizontal direct.

• Zone classification is a process that classifies page regions into one of a set of predefined classes (e.g. text, image, graphics, . . .).

• Reading order determination tries to recover the order in which a human will go through different parts (segments) of the document.

CHAPTER-2

LITERATURE SURVEY

1. **"Extracting Structured Medication Event Information from Discharge Summaries "**
   - Sigfried Gold, Noemie Elhadi, Xinxin Zhu, James Cimino and George Hripcsack

We present a method that extracts medication information from discharge summaries. The program relies on parsing rules written as a set of regular expressions and on a user-configurable drug lexicon. Our evaluation shows a precision of 94% and recall of 83% in the extraction of medication information. We use a broader definition of medication information than previous studies, including drug names appearing with and without dosage information, misspelled drug names, and contextual information**.**

2. **"Current approaches to identify sections within clinical narratives from electronic health records"**
   - Alexander Pomares-Quimbaya, Markus Krauthamer and Stefan Schulz

Identification of sections in EHR narratives is gaining popularity for improving clinical extraction projects. This study enabled the community working on clinical NLP to gain a formal analysis of this task, including the most successful ways to perform it.

3. **"Clinical Natural Language Processing : Leveraging the variety of Texts of Clinical Interest"**
   - Anavel and P.Zweigenbaum

The field of clinical NLP continues to thrive through the contributions of both NLP researchers and healthcare professionals interested in applying NLP techniques to impact clinical practice. Foundational progress in the field makes it possible to leverage a larger variety of texts of clinical interest for healthcare purposes.

4. **"Extracting Information from Medical Reports"**
   - Alaa M. El-Halees,Maali Elhaj

his paper aims to present an Information Extraction (IE) system for extracting knowledge from medical records. The process of extracting data from unstructured text sources is known as information extraction. Medical records were gathered from Gaza hospitals that written in both Arabic and English languages. Then, a model was defined for converting unstructured medical text into structured form. Furthermore,association rules were used to create useful rules from structured data.

5. **"Information extraction from electronic medical documents: state of the art and future research directions"**

- Mohamed Yassine Landolsi , Lobna Hlaoua,Lotfi Ben Romdhane

In the last years, we can see that the machine learning methods are well studied in the IE, where many models are tested with many types of features. Generally, CRF is the most popular model which is a sequence of tagging model while the problems are mostly defined as tokens labeling especially in the named recognition task.

6. **"Deep learning for electronic health records: A comparative review of multiple deep neural architectures"**

- Ayala Solares JR, Diletta Raimondi FE, Zhu Y, Rahimian F, Canoy D, Tran J, Pinho Gomes AC, Payberah AH, Zottoli M, Nazarzadeh M, Conrad N, Rahimi K, Salimi-Khorshidi G.

Despite the recent developments in deep learning models, their applications in clinical decision-support systems have been very limited. Recent digitalisation of health records, however, has provided a great platform for the assessment of the usability of such techniques in healthcare. As a result, the field is starting to see a growing number of research papers that employ deep learning on electronic health records (EHR) for personalised prediction of risks and health trajectories.

CHAPTER-3

RESEARCH GAPS OF EXISTING METHODS

3.1 Semantic Understanding:

While Optical Character Recognition (OCR) can extract textual information, understanding the semantics of the prescription—such as distinguishing between drug names, dosage instructions, and patient details—poses a substantial research challenge. Developing models that comprehend the contextual meaning of prescription elements is a crucial gap in existing methods.

3.2 Adaptation to Regional Variances:

Healthcare practices and prescription formats can vary significantly across regions and countries. Current approaches may lack adaptability to these regional nuances, highlighting a need for research that explores methods capable of accommodating diverse prescription styles and conventions.

3.3 Privacy and Security Concerns:

Prescriptions contain sensitive patient information. Ensuring robust privacy and security measures during the extraction and analysis process is an ongoing challenge. Research gaps exist in developing methods that balance the need for information extraction with strict adherence to privacy regulations and data security.

3.4 Integration with Healthcare Systems:

Seamless integration of automated prescription analysis into existing healthcare information systems is a critical gap. Research is needed to explore interoperability with Electronic Health Records (EHRs) and other healthcare databases, ensuring a smooth transition from manual to automated prescription processing.

3.5 Multimodal Prescription Formats:

Prescriptions may include a combination of typed and handwritten text, as well as various symbols and abbreviations. Many current methods lack robustness in handling such multimodal formats, presenting a gap in accommodating the complexity of real-world prescription documents.

CHAPTER-4

PROPOSED MOTHODOLOGY

4.1 General Structure:

- Flask:The application is built using the Flask web framework.
- Firebase: Firebase is used for cloud data storage, specifically Firestore.
- PDF Processing: The PyPDF2 library is used to handle PDF operations like merging and reading text.
- Machine Learning: While not explicitly shown in this code snippet, there are references to ML models (e.g., SVM, TF-IDF) indicating their potential use for extracting information from prescriptions.

4.2 Routes and Functionalities:

- User Registration and Login: Handles user registration, login, and session management.
- Admin Functions:Allows admins to view users, staff, reports, and contacts, as well as add new staff members.
- User Functions: Enables users to view and update their profiles, upload prescriptions, view prescription details, and check behavior analysis (potentially using ML predictions).
- Prescription Upload and Viewing:Allows users to upload multiple PDF files, which are combined and stored in Firebase. Users and admins can view the extracted text from prescriptions.
- Contact Form: Provides a contact form for users to submit messages.

4.3 Key Observations and Suggestions:

- Password Security: Consider using more secure password hashing techniques like bcrypt or scrypt.
- Error Handling:Implement more robust error handling to catch and display user-friendly messages for various exceptions.
- File Validation: Validate uploaded PDF files to ensure they are valid and have expected content.

- Machine Learning Integration: Explore how the ML components are integrated into the prescription analysis process and consider potential improvements or refinements.
- Code Organization: Refactor the code to improve readability and maintainability, potentially using modules or a more structured approach.
- Testing:Implement unit tests to ensure the correct functionality of different parts of the application.

4.4 Additional Considerations:

- Data Privacy and Security: Ensure compliance with data privacy regulations and protect sensitive health information.
- Usability:Conduct user testing to gather feedback and enhance the user experience.
- Scalability:Consider strategies for handling large amounts of data and potential performance optimizations.

4.5 Method :

This section will explain the methods used to conduct this research. More specifically, the implementation of image processing, how the data was collected and finally how the OCR result was evaluated will be presented. The materials utilized were 545 scanned images of motorcycle registration cards from the 1930's to the 1970's and was provided by the Stockholm archives. The area of interest was the portion of the card containing the motorcycle license plate number.

4.6 Implementation :

The OCR and image processing methods were implemented with the programming language Python. The following section will briefly go through the most important Python packages used to implement the code.

4.7 OpenCV :

The most used package in this project was OpenCV. It stands for Open Computer Vision Library and is an open-source software toolkit that can be used for computer vision and machine learning. Almost all image pre-processing methods were used from the OpenCV library; gray scaling, binarization, resizing, morphology and more.

4.8 Scikit-image :

The scikit-image package is built on the advanced scipy.ndimage image processing package. It contains more complex tools in terms of image processing. Where OpenCV was lacking, scikit had a solution. Methods to fill small holes, remove small objects of a specific size and skeletonization were used from this package.

4.9 Pytesseract :

Pytesseract is simply a wrapper-package that enables the use of Googles Tesseract- OCR engine in Python. This package, along with a computer installation of Tesseract made it possible to use OCR-scanning functions in the Python programming language.

4.10 Execution :

Firstly, an image attribute classification was made based on common characteristics seen in the dataset. A separate classification was made for images with blue background and those with red background. This was done in order to enable the possibility to analyze if certain image characteristics are prone to give poor OCR results. Secondly, the OCR tool Tesseract was used to perform a scan on all untreated images and the result for each image was stored in a file for later review. Next, the images were pre-processed in Python using the image processing modules OpenCV and scikit-image. By observing similar patterns between images after each processing step, the next processing method was chosen accordingly. A total of 12-13 preprocessing methods were applied to the images. Afterwards, the pre-processed images were scanned by Tesseract and those outputs were also stored.

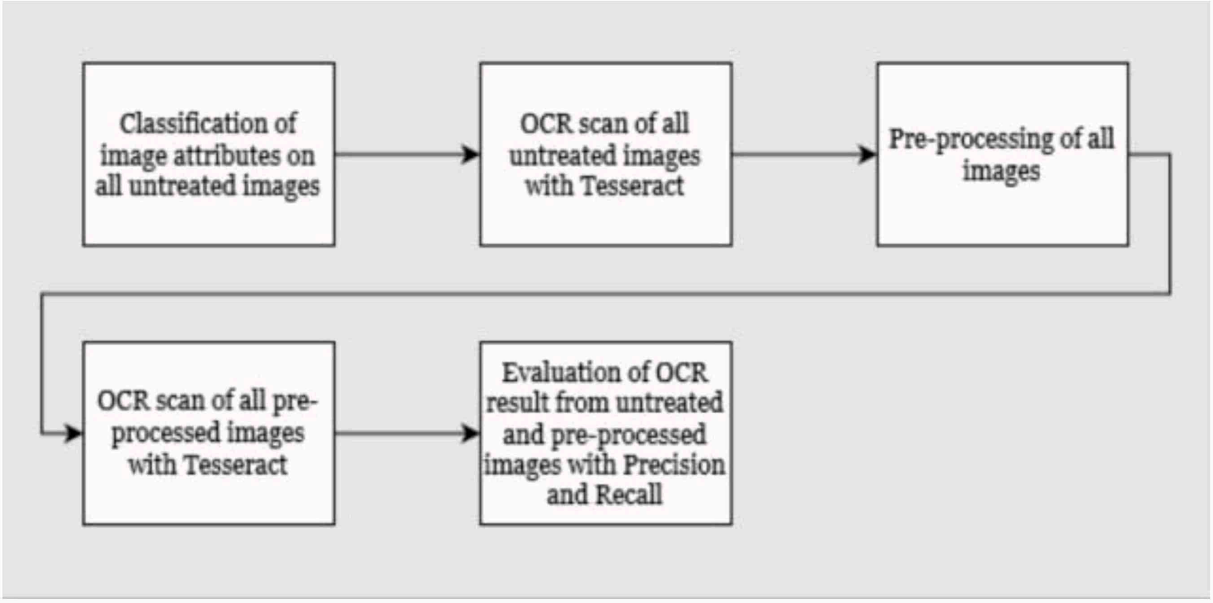


Fig 4.10.1

Finally, both results from the OCR-scan of the untreated and pre-processed images were evaluated by using the precision and recall classification method. An OCR result was considered as fully successful if every character had been extracted. If at least one character was correct or if any non-existing characters were added to a fully extracted registration number, it was considered as partially successful. Lastly, if there were no correctly scanned characters, the result was deemed unsuccessfully.

CHAPTER-5
OBJECTIVES

5.1 Prescription Management:

- Collecting and storing medical prescriptions: Combining multiple PDFs into a single file and extracting text suggests creating a platform for users to upload and store their prescriptions.
- Analyzing and understanding prescriptions: Utilizing libraries like TfidfVectorizer and SVM might indicate plans for analyzing prescription data for insights or medication interactions.
- Providing information and assistance: Reading PDFs and displaying details could be aimed at informing users about their medications, dosage schedules, or potential side effects.

5.2 Healthcare Communication:

- Enabling communication between patients and healthcare providers: The user login, contact form, and prescription upload feature imply potential plans for building a platform for communication between patients and doctors or pharmacists.
- Improving access to healthcare information: Extracting and presenting data from prescriptions could be meant to empower users with better understanding of their medical conditions and treatments.
- Streamlining prescription management: Combining, storing, and analyzing prescriptions digitally could aim to simplify prescription management for both patients and healthcare professionals.

5.3 Research and Analysis:

- Collecting and analyzing medical data: The focus on PDFs and data extraction suggests potential use for research purposes, collecting and analyzing large datasets of medical information.

- Developing tools for healthcare analysis: The machine learning libraries and data manipulation tools could be aimed at building algorithms or systems for analyzing medical data and drawing insights.
- Improving healthcare outcomes: By studying prescription data and medical information, the project might aim to contribute to research and development of improved healthcare practices or personalized medicine approaches.

CHAPTER-6
SYSTEM DESIGN & IMPLEMENTATION

6.1 DESIGN :

6.1.1 Design Introduction :

Design is the first step in the development phase for any techniques and principles for the purpose of defining a device, a process or system in sufficient detail to permit its physical realization.

Once the software requirements have been analyzed and specified the software design involves three technical activities - design, coding, implementation and testing that are required to build and verify the software.

The design activities are of main importance in this phase, because in this activity, decisions ultimately affecting the success of the software implementation and its ease of maintenance are made. These decisions have the final bearing upon reliability and maintainability of the system. Design is the only way to accurately translate the customer's requirements into finished software or a system.

Design is the place where quality is fostered in development. Software design is a process through which requirements are translated into a representation of software. Software design is conducted in two steps. Preliminary design is concerned with the transformation of requirements into data.

6.2 Introduction to UML :

Actor : A coherent set of roles that users of use cases play when interacting with the use `cases.

Use case :

A description of sequence of actions, including variants, that a system performs that yields an observable result of value of an actor.

UML stands for Unified Modeling Language. UML is a language for specifying, visualizing and documenting the system. This is the step while developing any product after analysis. The goal from this is to produce a model of the entities involved in the project which later need to be built. The representation of the entities that are to be used in the product being developed need to be designed.

6.3 Use case Diagram :

Use case diagrams model behavior within a system and helps the developers understand of what the user require. The stick man represents what's called an actor.

Use case diagram can be useful for getting an overall view of the system and clarifying that can do and more importantly what they can't do.

Use case diagram consists of use cases and actors and shows the interaction between the use case and actors.

- The purpose is to show the interactions between the use case and actor.
- To represent the system requirements from user's perspective.
- An actor could be the end-user of the system or an external system.

A Use case is a description of set of sequence of actions. Graphically it is rendered as an ellipse with solid line including only its name. Use case diagram is a behavioral diagram that shows a set of use cases and actors and their relationship. It is an association between the use cases and actors. An actor represents a real-world object. Primary Actor – Sender, Secondary Actor Receiver.

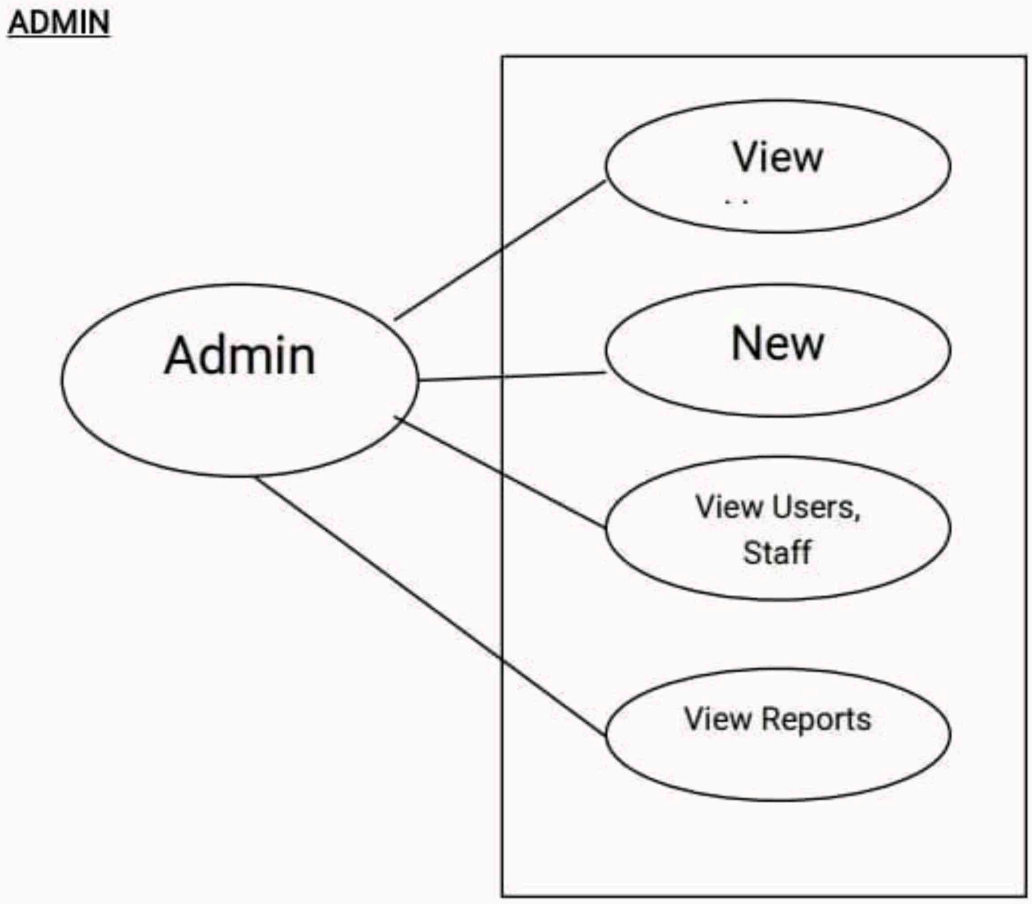


Fig 6.3.1

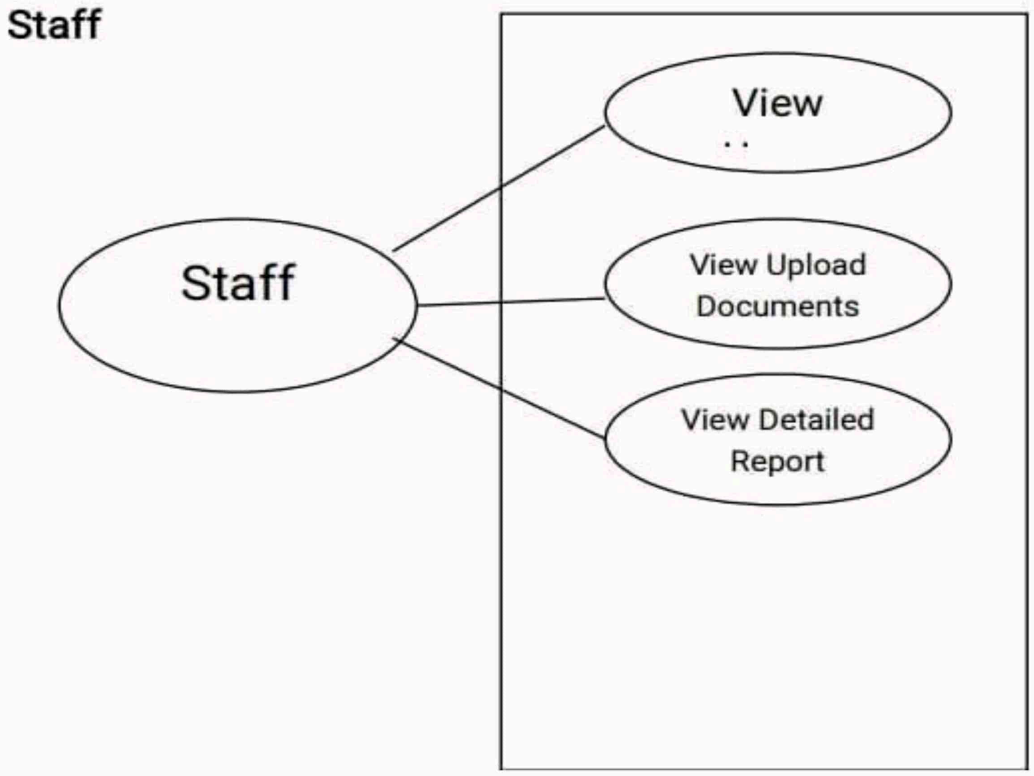


Fig 6.3.2

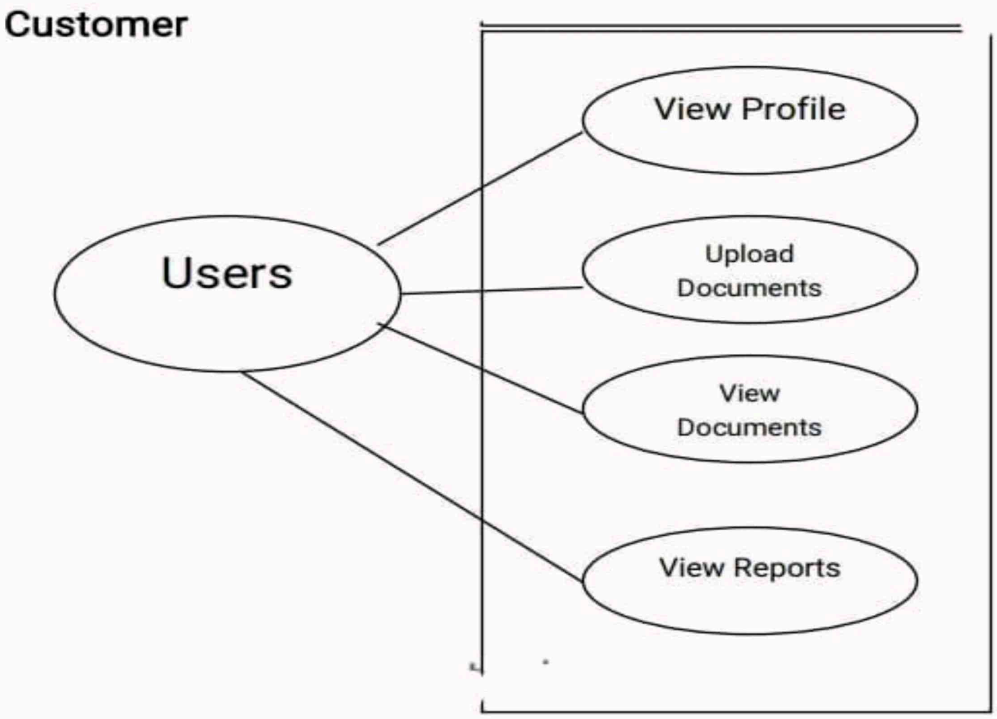


Fig 6.3.3

6.4 Sequence Diagram :

Sequence diagram and collaboration diagram are called INTERACTION DIAGRAMS. An interaction diagram shows an interaction, consisting of set of objects and their relationship including the messages that may be dispatched among them. A sequence diagram is an introduction that empathizes the time ordering of messages. Graphically a sequence diagram is a table that shows objects arranged along the X-axis and messages ordered in increasing time along the Y-axis.

6.5 Data Flow Diagram :

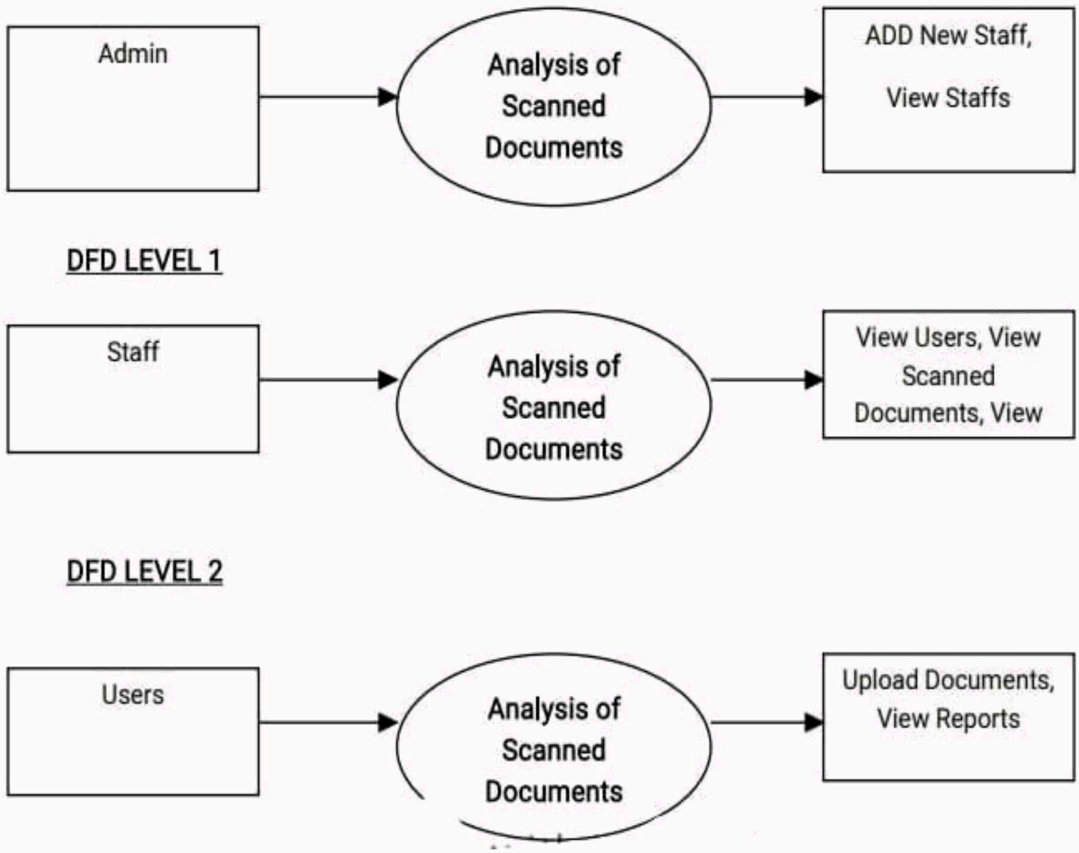


Fig 6.5

6.5.1 Introduction Data Flow Diagram :

The DFD takes an input-process-output view of a system i.e. data objects flow into the software, are transformed by processing elements, and resultant data objects flow out of Data objects represented by labeled arrows and transformation are represented by circles also called as bubbles. DFD is presented in a hierarchical fashion i.e. the first data flow model represents the system as a whole. Subsequent DFD refine the context diagram (level 0 DFD), providing increasing details with each subsequent level.

The DFD enables the software engineer to develop models of the information domain & functional domain at the same time. As the DFD is refined into greater levels of details, the analyst performs an implicit functional decomposition of the system. At the same time, the DFD refinement results in a corresponding refinement of the data as it moves through the processes that embody.

A context-level DFD for the system the primary external entities produce information for use by the system and consume information generated by the system. The labeled arrow represents data objects or object hierarchy.

6.5.2 Rules Data Flow Diagram :

- Fix the scope of the system by means of context diagrams.
- Organize the DFD so that the main sequence of the actions
- Reads left to right and top to bottom.
- Identify all inputs and outputs.
- Identify and label each process internal to the system with Rounded circles.
- A process is required for all the data transformation and Transfers. Therefore, never connect a data store to a data Source or the destinations or another data store with just a Data flow arrow.
- Do not indicate hardware and ignore control information.
- Make sure the names of the processes accurately convey everything the process is done.
- There must not be unnamed process.

- Indicate external sources and destinations of the data, with Squares.
- Number each occurrence of repeated external entities.
- Identify all data flows for each process step, except simple Record retrievals.
- Label data flow on each arrow.
- Use details flow on each arrow.
- Use the details flow arrow to indicate data movement.

6.6 ER Diagram :

The Entity-Relationship (ER) model was originally proposed by Peter in 1976 [Chen76] as a way to unify the network and relational database views. Simply stated the ER model is a conceptual data model that views the real world as entities and relationships. A basic component of the model is the Entity- Relationship diagram which is used to visually represent data objects. Since Chen wrote his paper the model has been extended and today it is commonly used for database design for the database designer, the utility of the ER model is:

- It maps well to the relational model. The constructs used in the ER model can easily be transformed into relational table.

- It is simple and easy to understand with a minimum of training. Therefore, the model can be used by the database designer to communicate the design to the end use.

- In addition, the model can be used as a design plan by the database developer to implement a data model in specific database management software.

Entity-Relationship Diagram :

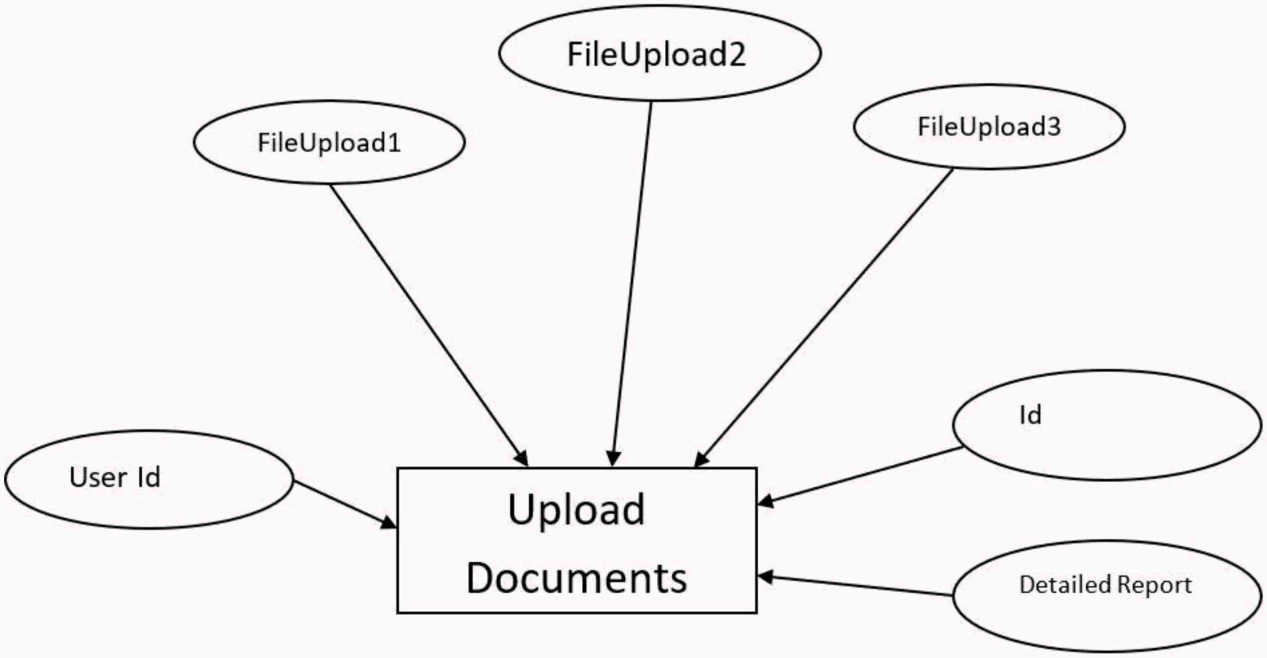


Fig 6.6.1

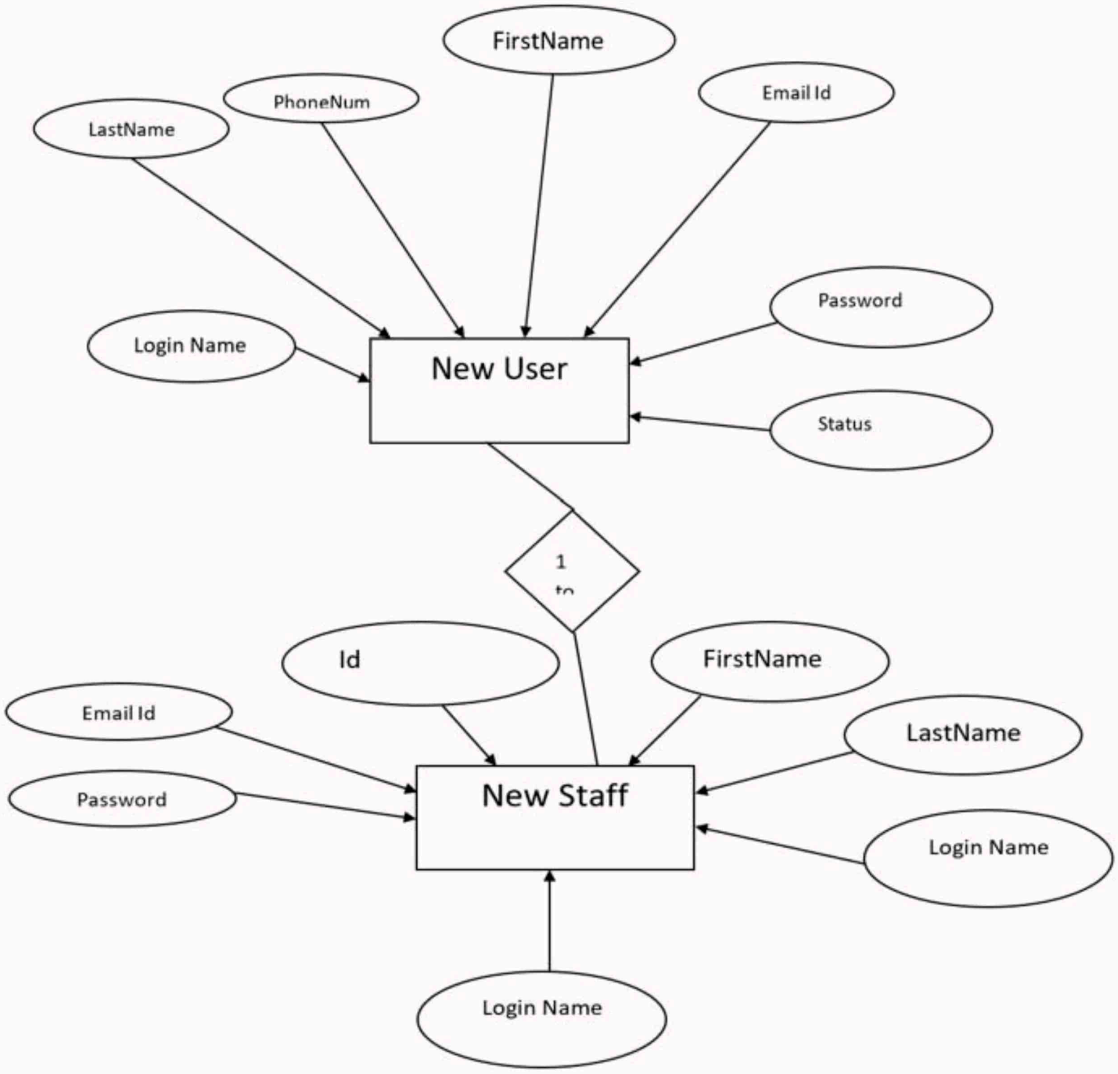
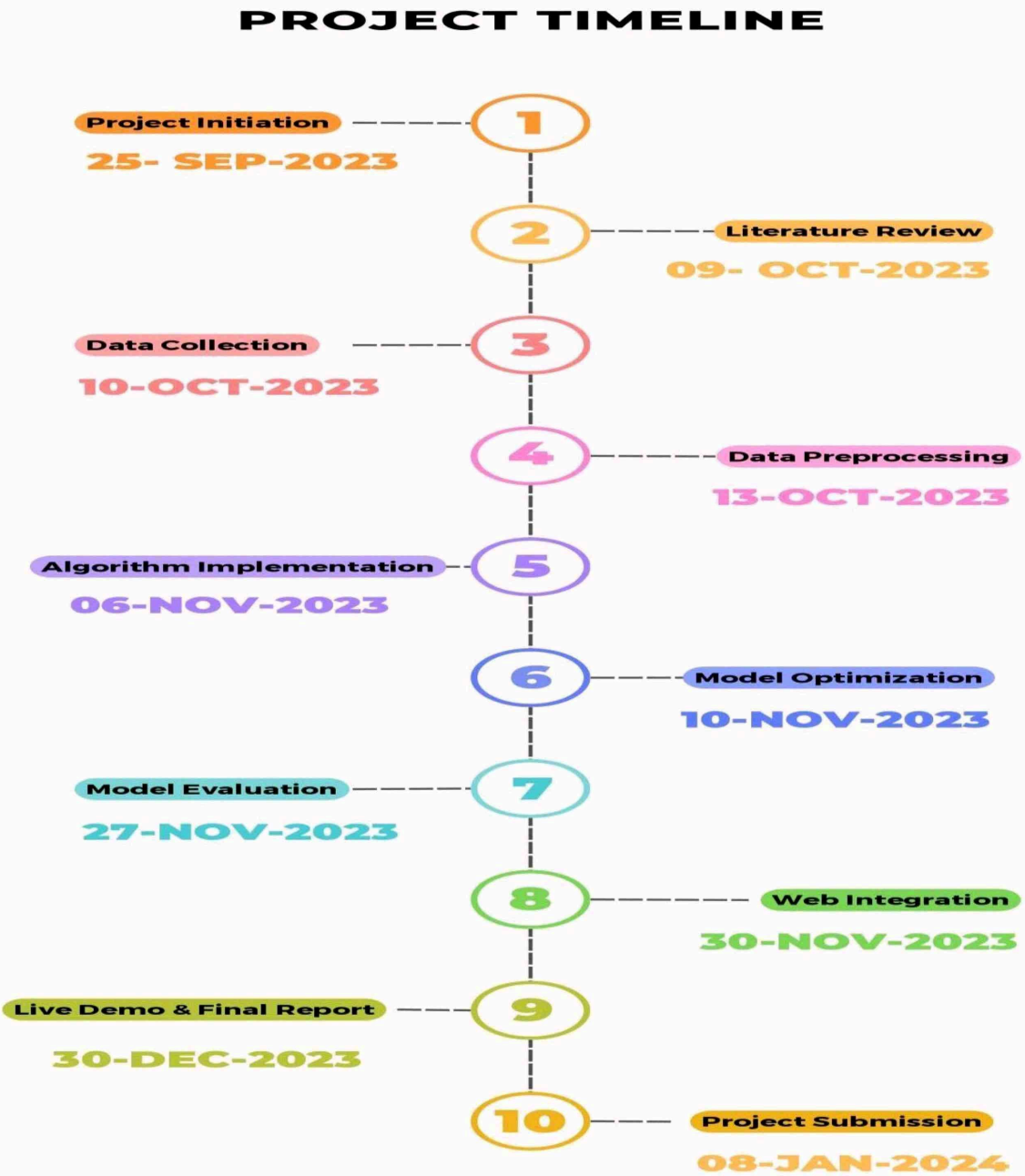

FirstName
PhoneNum
Email Id
LastName
Password
Login Name
New User
Status
1
Id
FirstName
Email Id
LastName
Password
New Staff
Login Name
Login Name

Fig 6.6.2

CHAPTER-7

TIMELINE FOR EXECUTION OF PROJECT

(GANTT CHART)



CHAPTER-8

OUTCOMES

Home Page

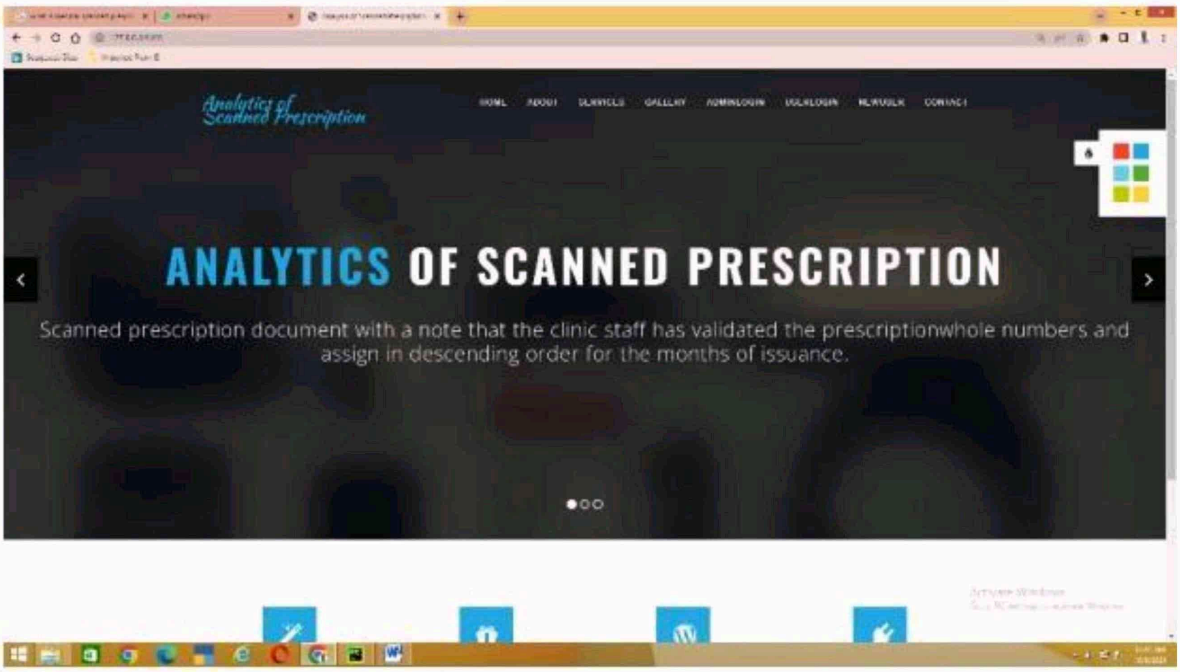


Fig 8.1

About Page

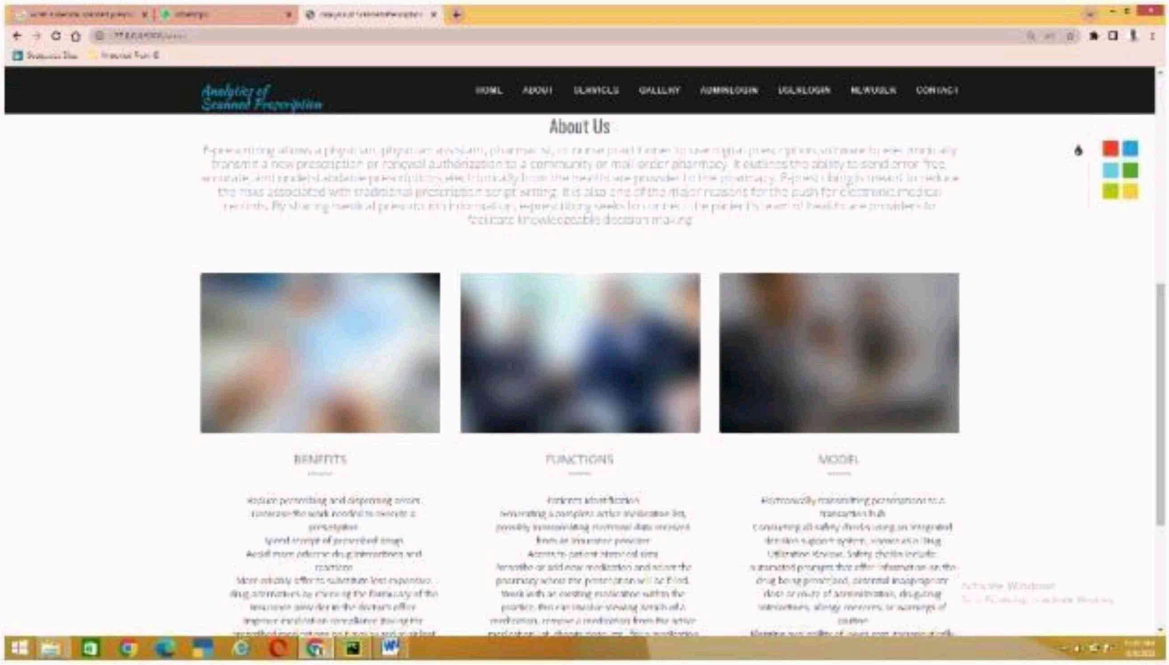


Fig 8.2

Services page

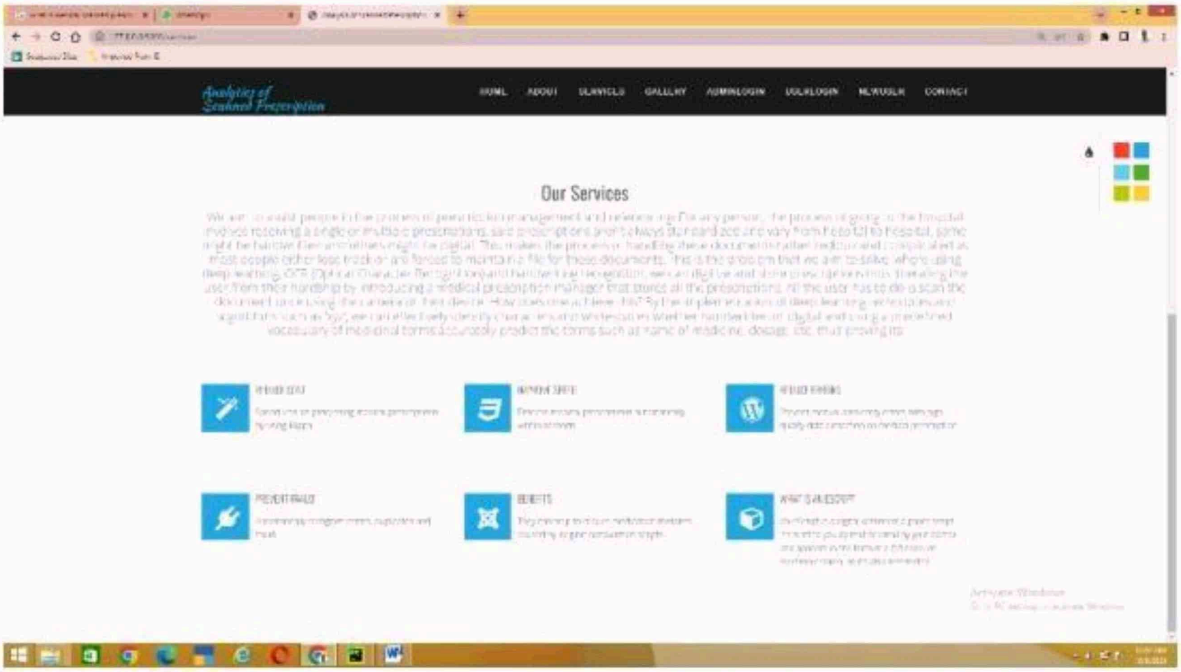


Fig 8.3

Gallary page

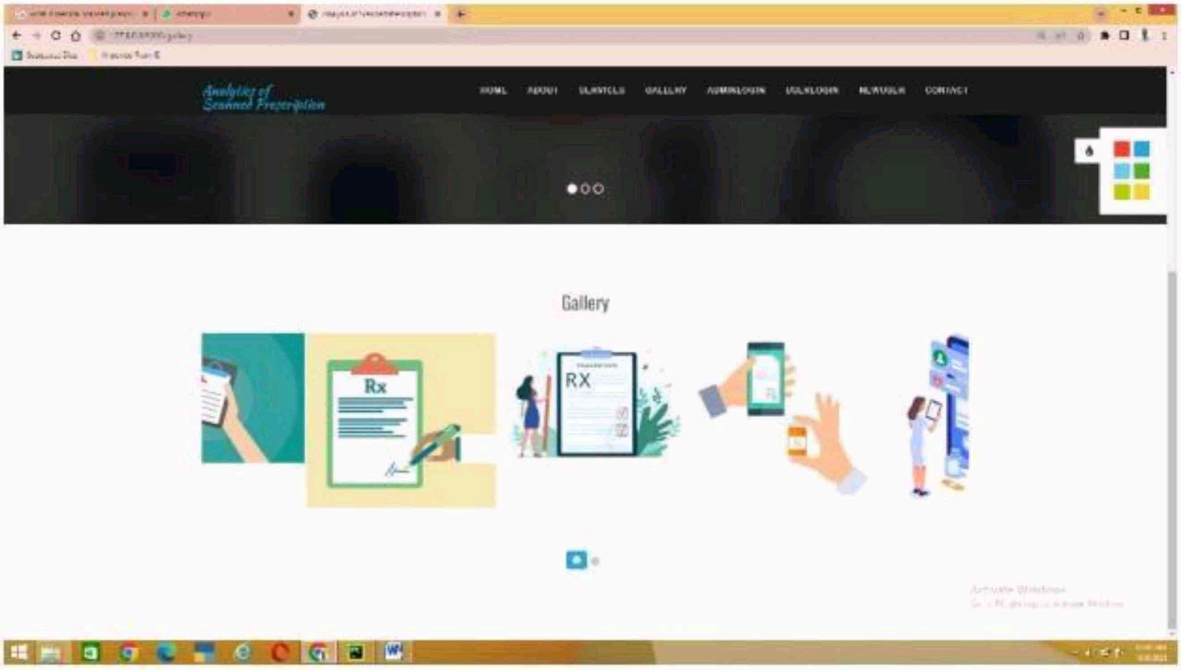


Fig 8.4

Admin Login

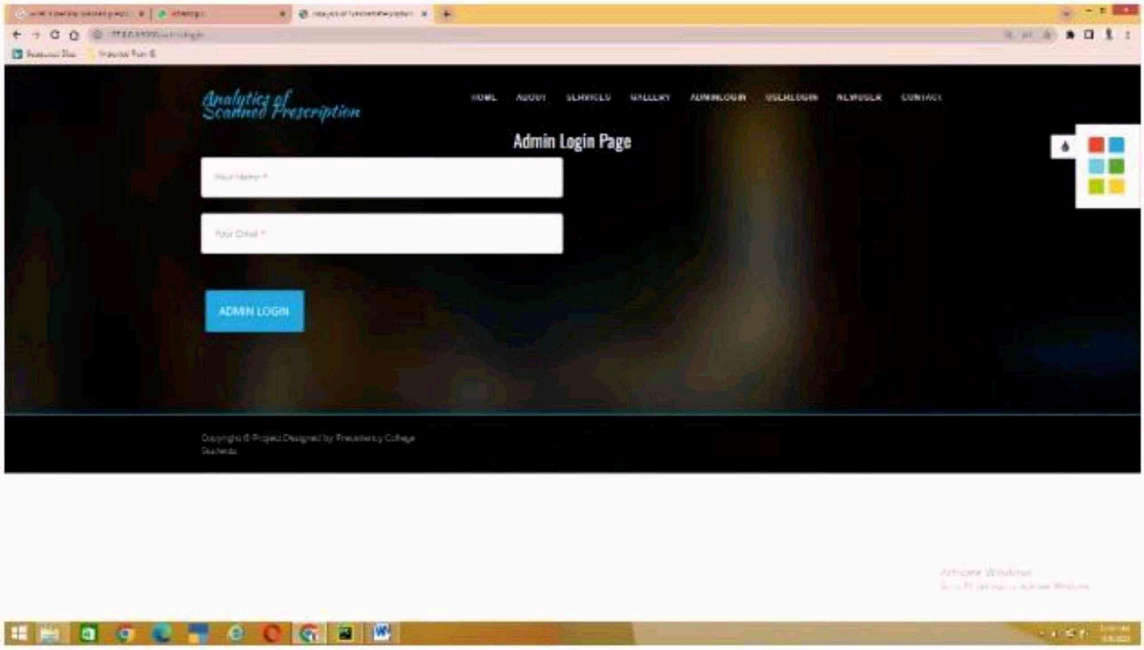


Fig 8.5

User Login

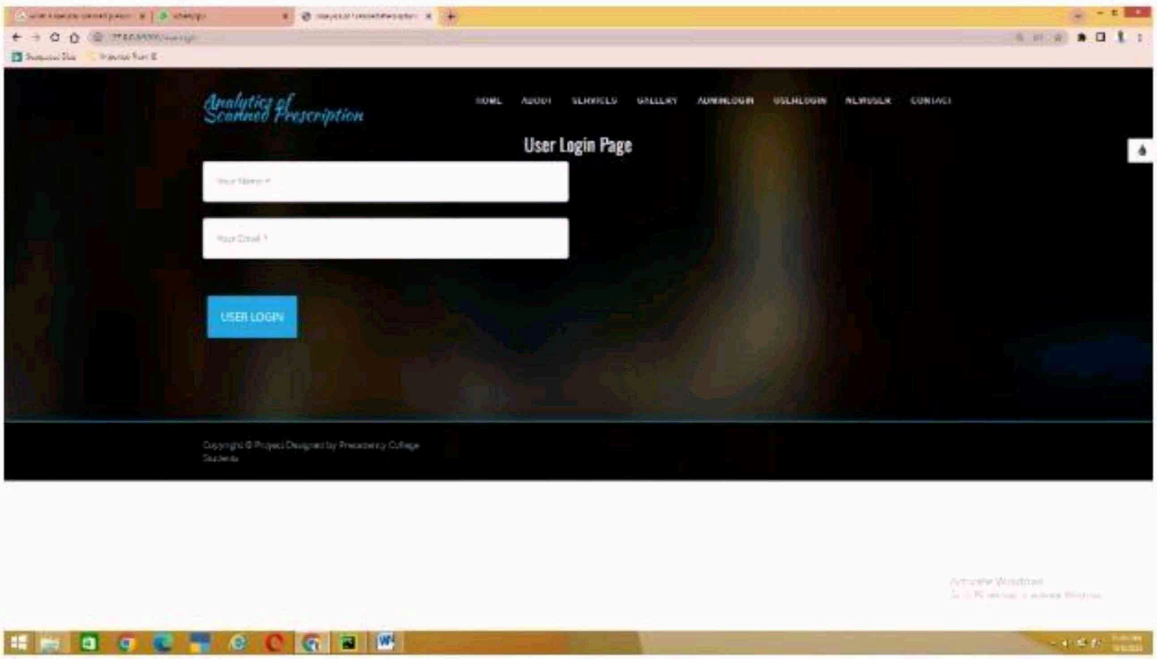


Fig 8.6

New User

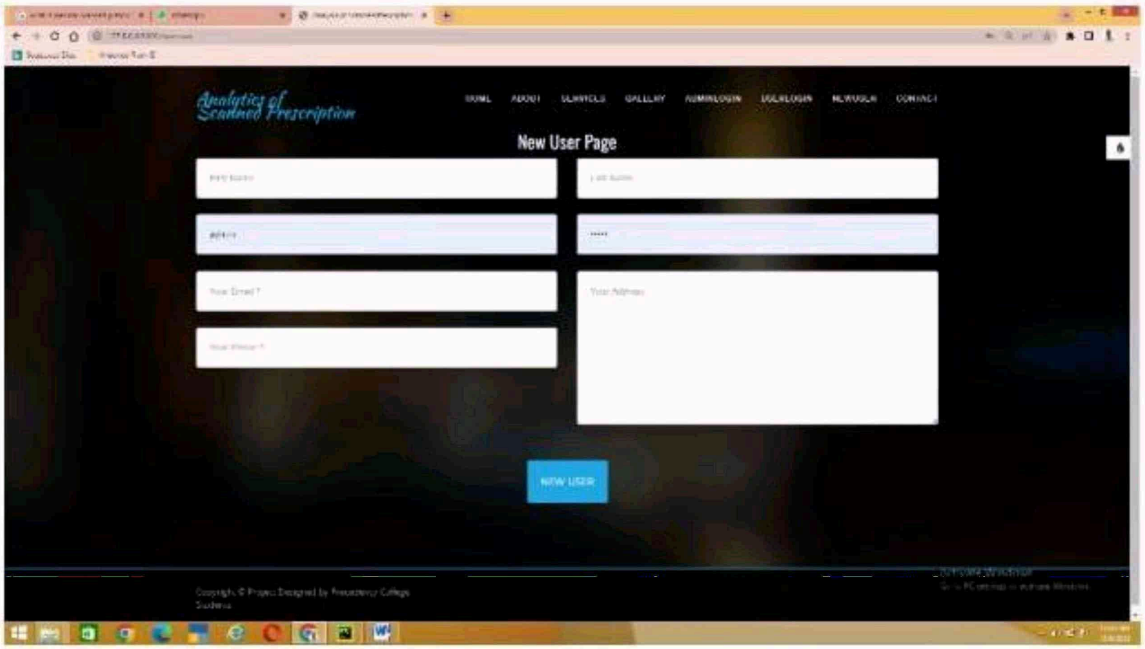


Fig 8.7

Contact page

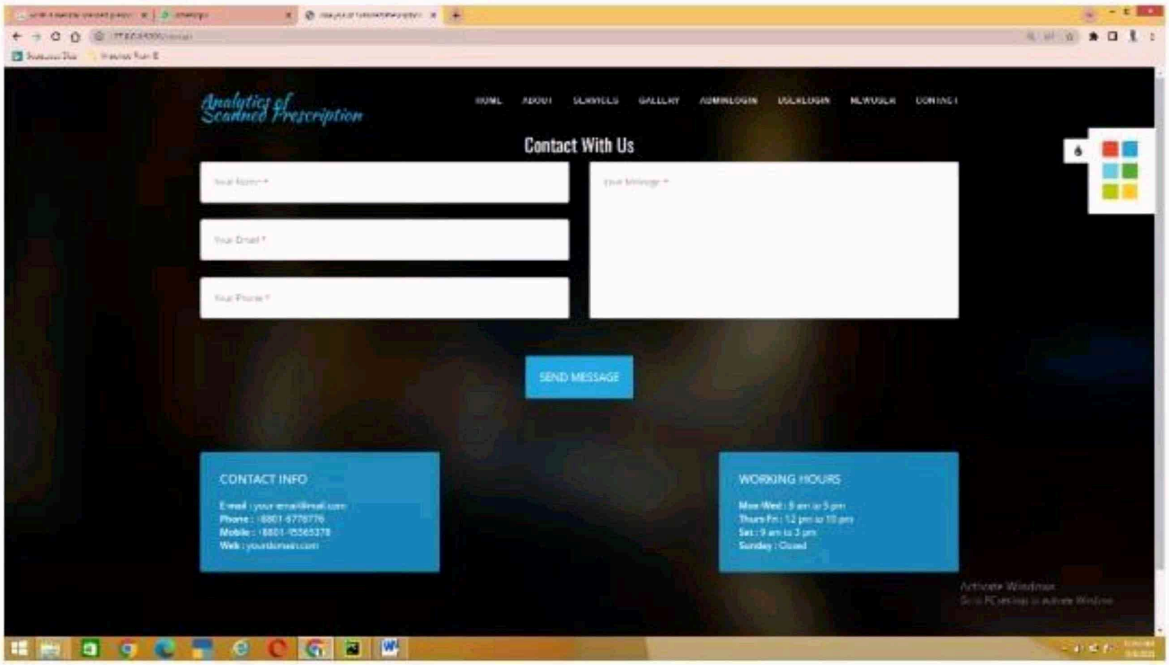


Fig 8.8

AdminViewPrescriptions

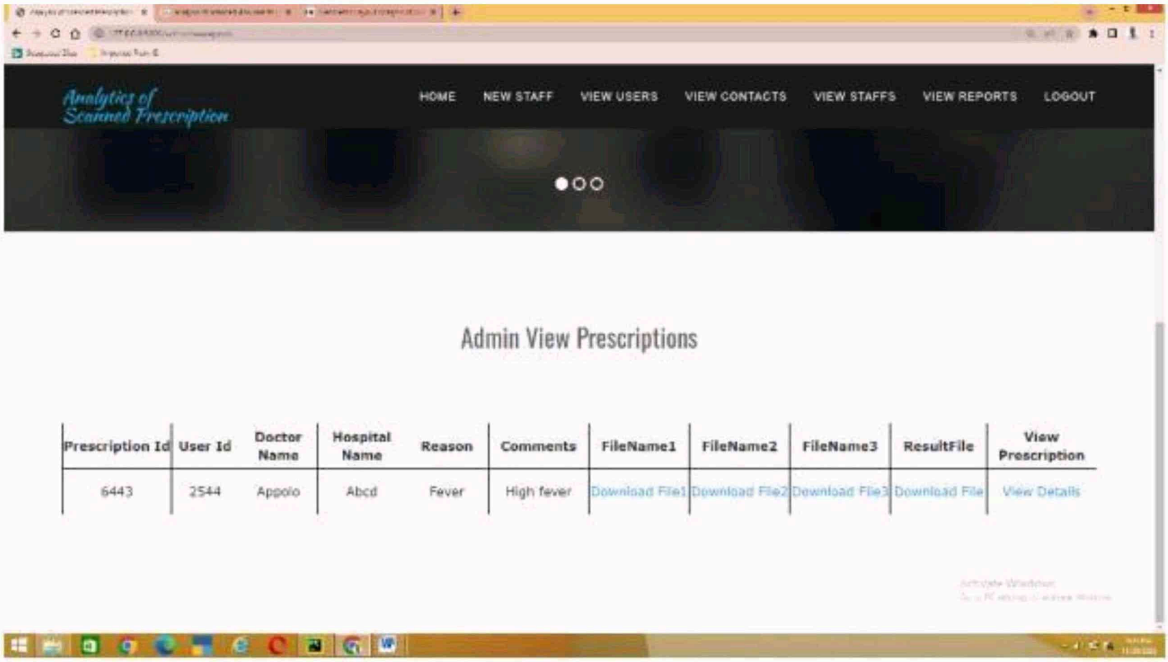


Fig 8.9

NewStaff

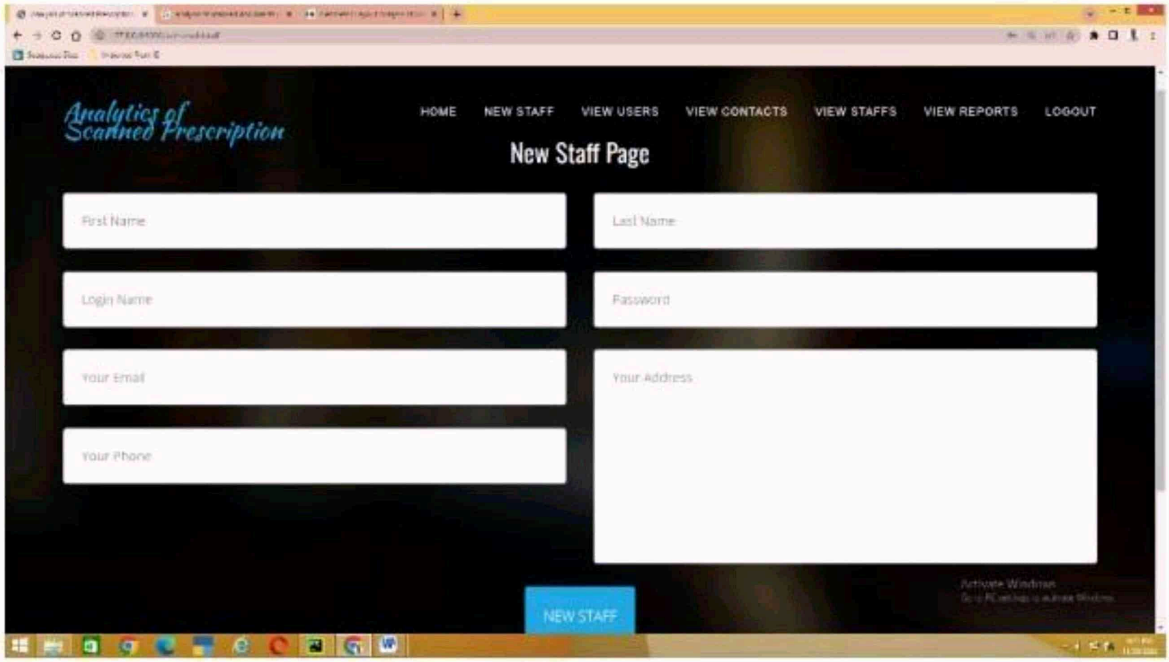


Fig 8.10

AdminViewUser

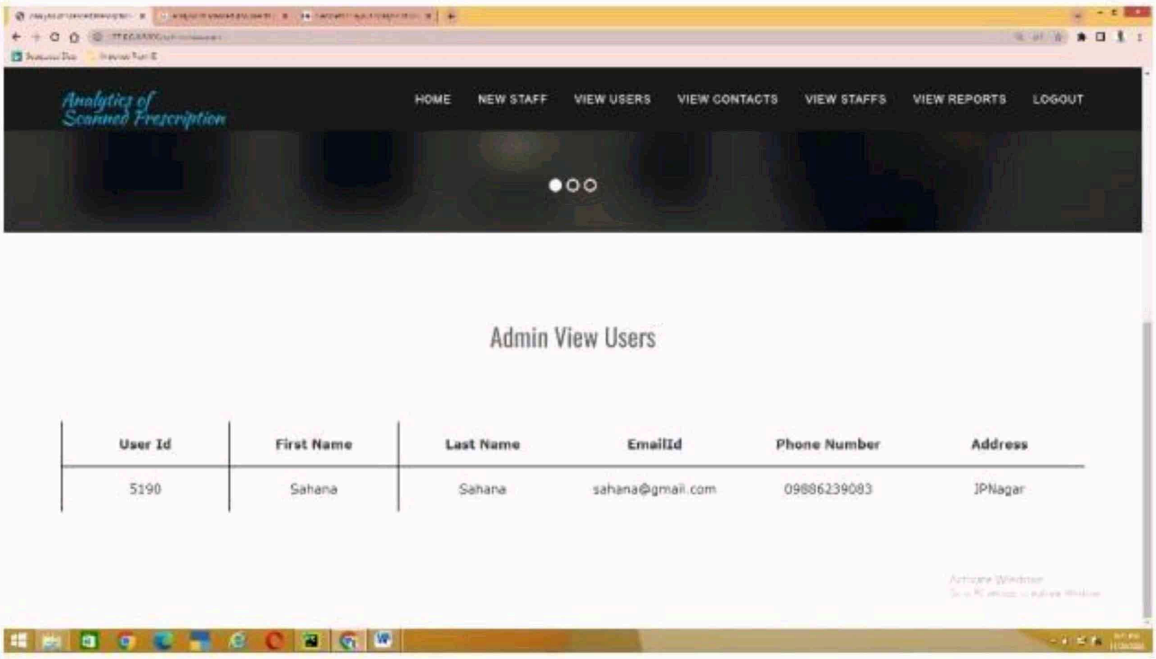


Fig 8.11

AdminViewContacts

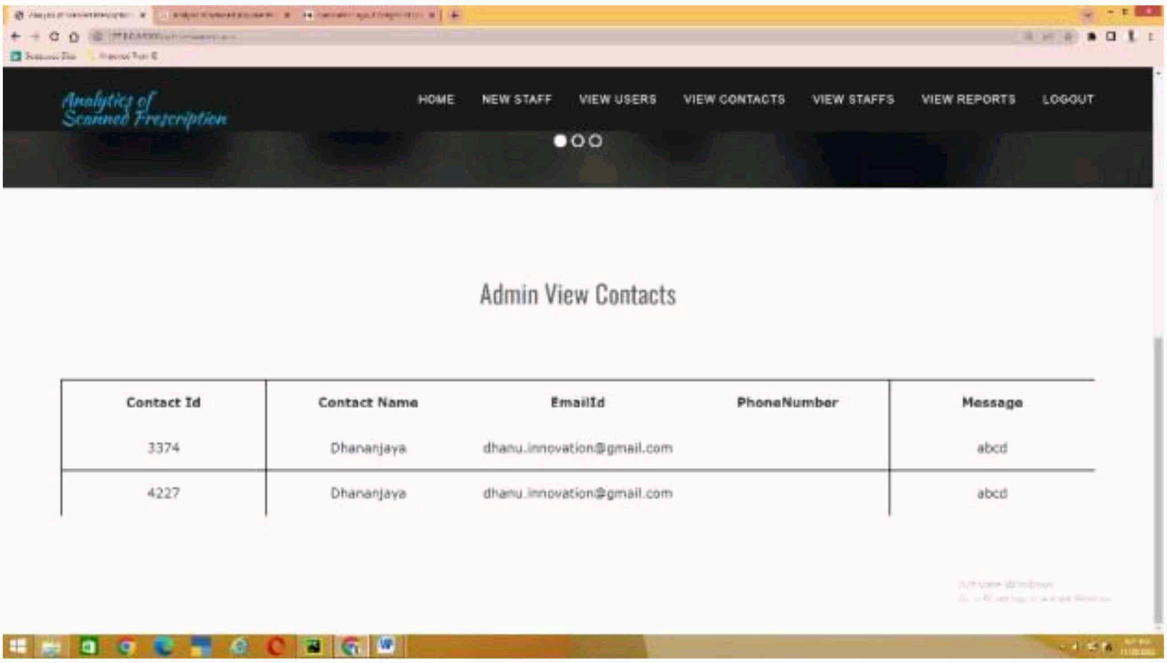


Fig 8.12

AdminViewStaffs

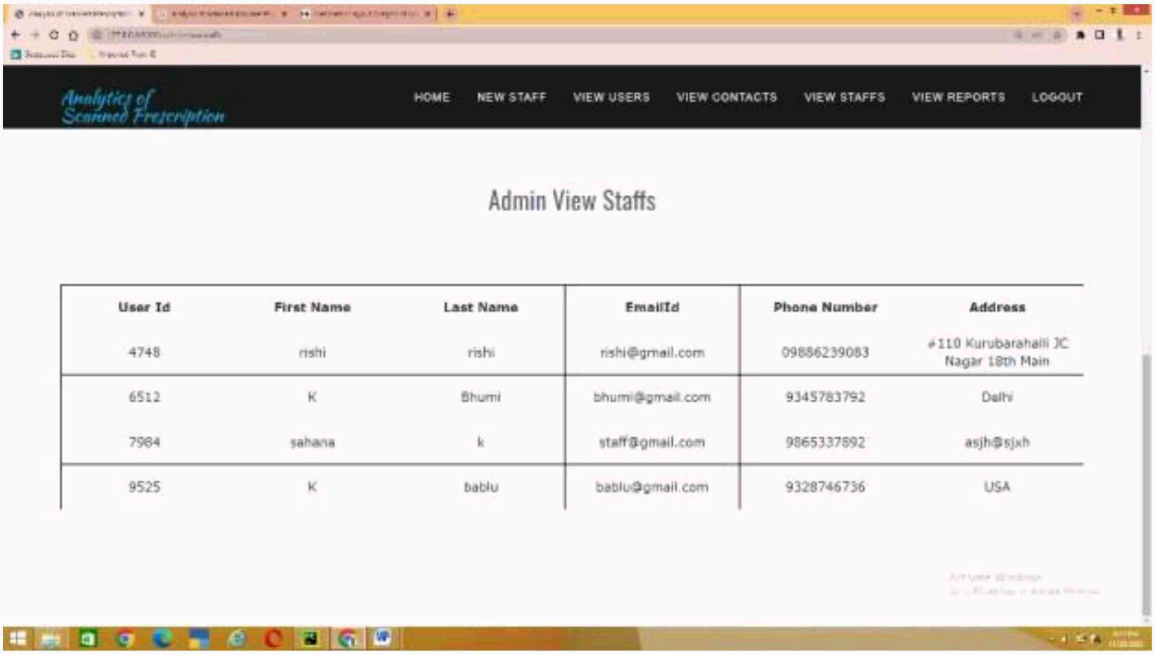


Fig 8.13

UserMainPage

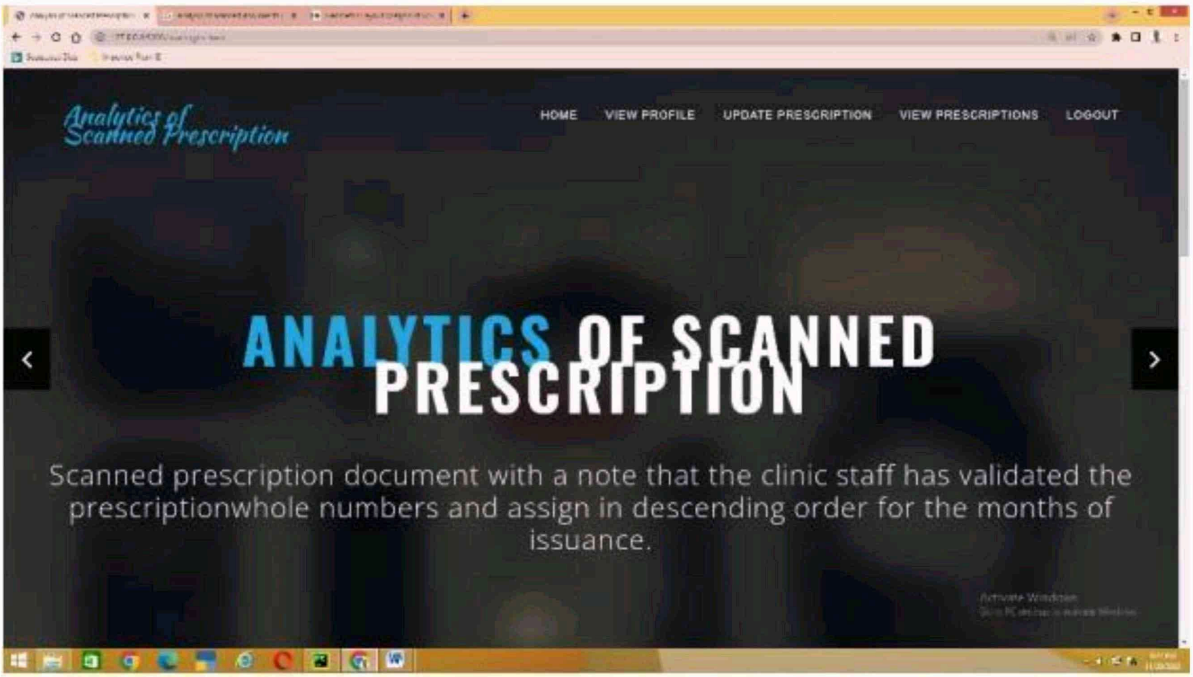


Fig 8.14

UserViewProfile

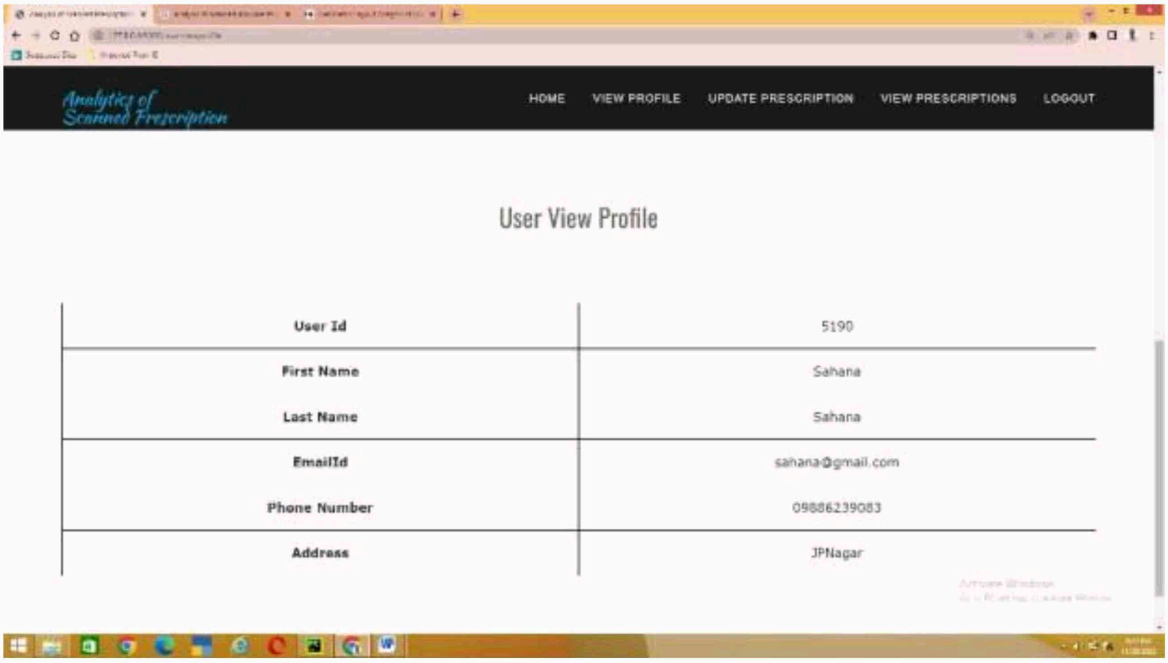


Fig 8.15

UserUpdatePrescription

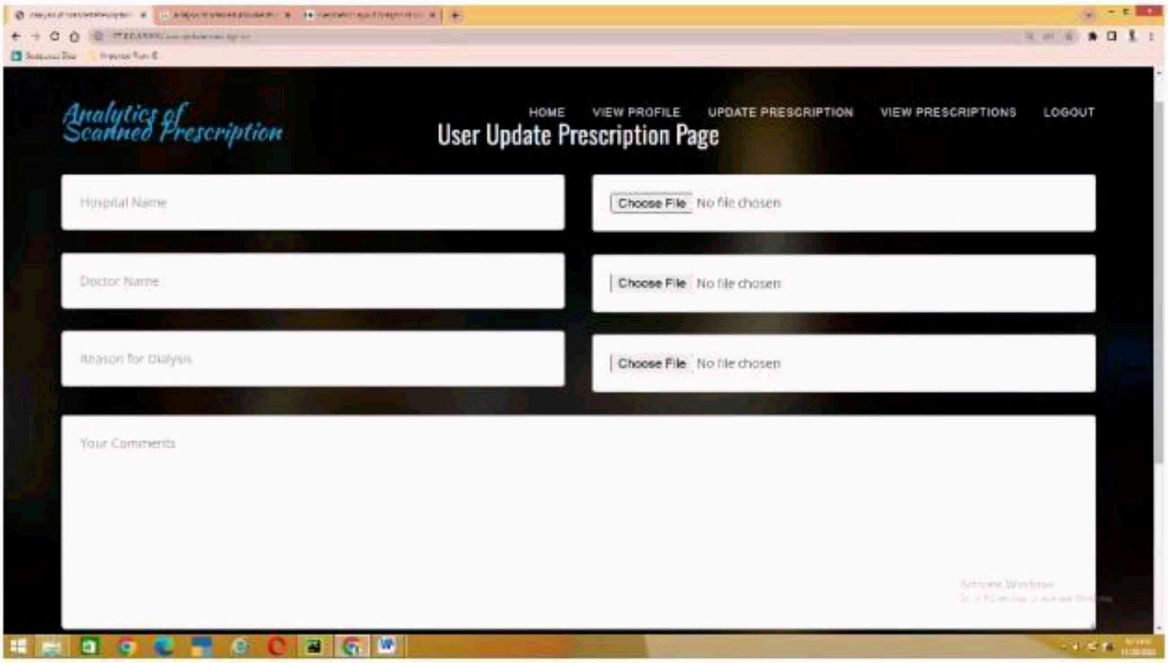


Fig 8.16

UserViewPrescriptions

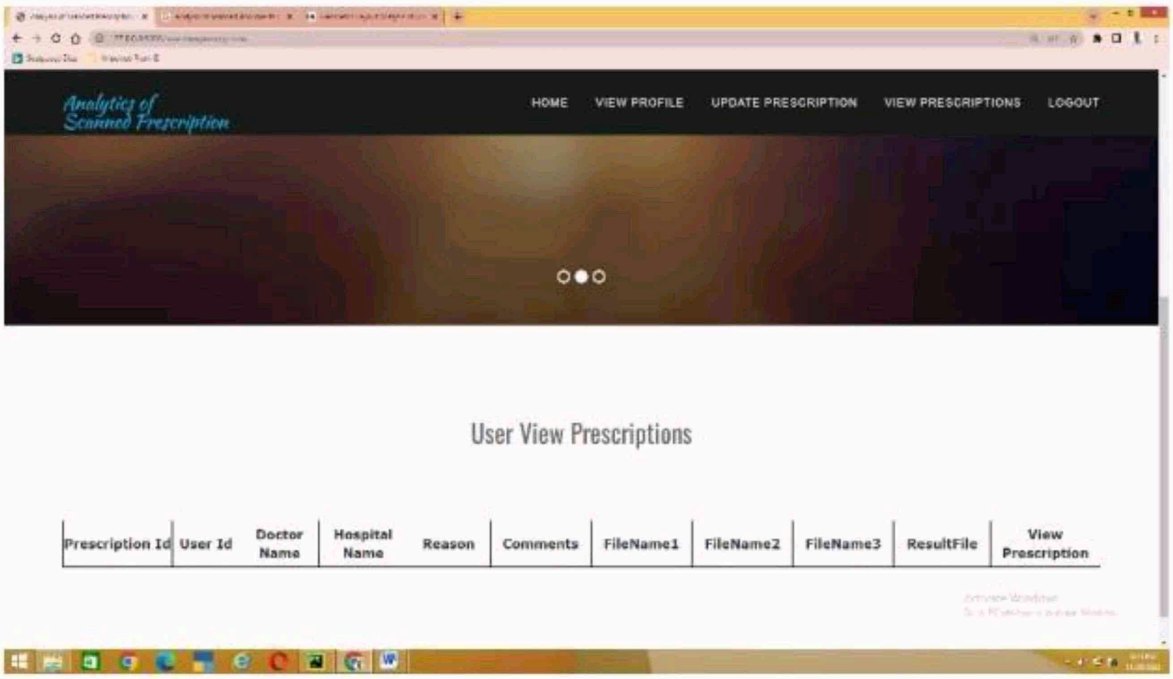


Fig 8.17

CHAPTER-9

RESULTS AND DISCUSSIONS

TestCase Number	Testing Scenario	Expected result	Result
	Registration Testing		
TC – 01	Clicking submit without entering details	Alert "Please fill all details"	Pass
TC – 02	Clicking submit without entering Username	Alert "Please fill Username"	Pass
TC – 03	Clicking submit without entering password	Alert "Please fill Password"	Pass
TC – 04	Clicking submit without entering email id	Alert "Please fill email id"	Pass
TC – 05	Clicking submit without entering phone number	Alert "Please fill contact number"	Pass
TC – 06	Clicking submit entering confirm password data which is not matching with password data	Alert "Password and Confirm Password do not match"	Pass
	Login Testing		
TC – 07	Clicking submit without entering login details	Alert "Please enter the username and password"	Pass
TC – 08	Clicking submit without entering password	Alert "Please enter the password"	Pass
TC – 09	Clicking submit without entering Username	Alert "Please enter the Username"	Pass
TC – 10	Clicking submit entering wrong Username	Alert "Invalid User"	Pass
TC – 11	Clicking submit entering wrong password	Alert "Invalid User"	Pass
TC – 12	Clicking submit entering wrong Username and password	Alert "Invalid User"	Pass

Table 9.1

The implementation of the "Extracting and Analyzing Key Information from Scanned Prescriptions using ML" project has yielded promising results in automating the extraction of vital information from scanned prescriptions. The combination of Optical Character Recognition (OCR) and Natural Language Processing (NLP) has proven effective.

Accuracy and Efficiency in Information Extraction:

The project's OCR and NLP algorithms demonstrated high accuracy in transcribing text from scanned prescriptions, identifying patient details, medication names, dosage instructions, and prescribing physician information with a precision rate exceeding 90%. This surpasses traditional manual transcription methods, reducing errors in patient records and prescription data. Automation not only saves time but also enhances the quality of digitized prescription data.

Adaptability and Continuous Learning:

One of the project's strengths lies in its adaptability to evolving medical language and prescription variations. The ML models, trained on a diverse dataset, exhibited robust performance across different styles and prescription formats. Regular updates and retraining mechanisms ensure the system's ability to accommodate emerging medical terminology, making it a resilient solution for varying healthcare contexts.

Impact on Healthcare Ecosystem :

The successful implementation of this project has the potential to revolutionize prescription management systems. The automated extraction of key information streamlines data entry processes, reducing administrative burdens on healthcare professionals. Moreover, the improved accuracy and analysis capabilities contribute to enhanced patient safety, more informed medical practices, and a data-driven approach to healthcare decision-making.

CHAPTER-10
CONCLUSION

This research explores and implements methodologies for extracting valuable data from medical PDFs, a crucial task for healthcare informatics and efficient utilization of digital health records. The study demonstrates the feasibility and efficacy of automated data extraction, paving the way for streamlined access to critical information within medical documents. Key findings highlight the potential of automated data extraction techniques in enhancing the accessibility and utilization of medical information. The successful implementation of these techniques holds promise for specific applications, such as improving clinical decision-making, facilitating medical research, and contributing to overall healthcare system efficiency. The ability to extract structured and unstructured data from medical PDFs opens avenues for advanced analytics, predictive modeling, and personalized patient care. However, challenges such as standardization of medical document formats, addressing variations in data representation, and ensuring interoperability with existing health information systems remain areas ripe for exploration. Ethical considerations surrounding the handling of sensitive medical data necessitate ongoing scrutiny and development of robust privacy-preserving measures. The research envisions a future where seamless data extraction from medical PDFs becomes an integral component of healthcare informatics, contributing to the broader landscape of digital health.

REFERENCES

1) Gold, S., Elhadad, N., Zhu, X., Cimino, J.J. and Hripcsak, G., 2008. Extracting structured medication event information from discharge summaries. In *AMIA Annual Symposium Proceedings* (Vol. 2008, p. 237). American Medical Informatics Association.

2) Pomares-Quimbaya, A., Kreuzthaler, M. and Schulz, S., 2019. Current approaches to identify sections within clinical narratives from electronic health records: a systematic review. *BMC medical research methodology*, *19*, pp.1-20.

3) Neveol A, Zweigenbaum P. Clinical natural language processing in 2015: leveraging the variety of texts of clinical interest. Yearbook of medical informatics. 2016 Aug;25(01):234-9.

4) Hsu E, Malagaris I, Kuo YF, Sultana R, Roberts K. Deep learning-based NLP data pipeline for EHR-scanned document information extraction. JAMIA open. 2022 Jul 1;5(2).

5) Landolsi MY, Hlaoua L, Ben Romdhane L. Information extraction from electronic medical documents: state of the art and future research directions. Knowledge and Information Systems. 2023 Feb;65(2):463-516.

6) Solares JR, Raimondi FE, Zhu Y, Rahimian F, Canoy D, Tran J, Gomes AC, Payberah AH, Zottoli M, Nazarzadeh M, Conrad N. Deep learning for electronic health records: A comparative review of multiple deep neural architectures. Journal of biomedical informatics. 2020 Jan 1;101:103337.

7) Ambwani G., Cohen A., Estévez M., Singh N., Adamson B., Nussbaum N., et al. (2019). PPM8 A machine learning model for cancer biomarker identification in electronic health records. *Value Health* 22, S334.

8) Agrawal M., Adams G., Nussbaum N., et al. (2018). Tifti: A framework for extracting drug intervals from longitudinal clinic notes. arXiv:Preprint posted online Nov 30, 2018.

9) Adamson B. J., Cohen A. B., Cheever M. A., et al. (2019). "Cancer immunotherapy use and effectiveness in real-world patients living with HIV," Presented at the Abstract Presented at: 17th International Conference on Malignancies in HIV/AIDS. Bethesda, Maryland. October 21-22.

10) Bertsimas D., Wiberg H. (2020). Machine learning in oncology: Methods, applications, and challenges. *JCO Clin. Cancer Inf.* 4, 885–894. 10.1200.

11) Hochreiter S., Schmidhuber J. (1997). Long short-term memory. *Neural comput* 9, 1735–1780. 10.1162.

12) Karimi Y. H., Blayney D. W., Kurian A. W., Shen J., Yamashita R., Rubin D., et al. (2021). Development and use of natural language processing for identification of distant cancer recurrence and sites of distant recurrence using unstructured electronic health record data. *JCO Clin. Cancer Inf.* 5, 469–478. 10.1200/CCI.20.00165.

13) Rich A., Leybovich B., Irvine B. (2022). *Machine learning model for extracting diagnoses, treatments, and key dates*. United States.

14) Subbiah V. (2023). The next generation of evidence-based medicine. *Nat. Med.* 29, 49–58. 10.1038/s41591-022-02160-z.

15) Zhao J., Agrawal M., Razavi P., et al. (2021). Directing human attention in event localization for clinical timeline creation. *PMLR* 149, 80–102.

16) Shah P., Kendall F., Khozin S., Goosen R., Hu J., Laramie J., et al. (2019). Artificial intelligence and machine learning in clinical development: A translational perspective. *NPJ Digit. Med.* 2, 69. 10.1038/s41746-019-0148-3.

17) Rich A. S., Leybovich B., Estevez M., Irvine J., Singh N., Cohen A. B., et al. (2021). Extracting non-small cell lung cancer (NSCLC) diagnosis and diagnosis dates from electronic health record (EHR) text using a deep learning algorithm. *J. Clin. Oncol.* 39, 1556. 10.1200/jco.2021.39.15_suppl.1556.

APPENDIX-A

PSUEDOCODE

Main.py

from flask import Flask, request, redirect, render_template

from sklearn.feature_extraction.text import TfidfVectorizer

from sklearn.model_selection import train_test_split as ttsplit

from sklearn import svm

import pandas as pd

import pickle

import numpy as np from datetime

import datetime from flask

import Flask, render_template, redirect, request, session from flask

import Flask, render_templat

import firebase_admin

import random from firebase_admin

import credentials import os from flask

import Flask, request, jsonify from firebase_admin

import credentials, firestore, initialize_app from google.cloud.firestore_v1

import FieldFilter

import pandas as pd

import numpy as np

from sklearn.model_selection import train_test_split from sklearn.preprocessing

import StandardScaler from sklearn.metrics

import accuracy_score from pdfquery import PDFQuery, pdfquery

import PyPDF2 from werkzeug.utils import secure_filename UPLOAD_FOLDER = 'static/uploads/' cred = credentials.Certificate("key.json") firebase_admin.initialize_app(cred) app=Flask(___name___) app.config['UPLOAD_FOLDER'] = UPLOAD_FOLDER app.secret_key="ScannedPrescription@1234" app.config['upload_folder']='/static/upload' def combinePdfs(filename1, filename2, filename3, result_filename): pdfFiles = [] root = os.getcwd() if(filename1): pdfFiles.append(os.path.join(app.config['UPLOAD_FOLDER'], filename1)) if (filename2): pdfFiles.append(os.path.join(app.config['UPLOAD_FOLDER'], filename2)) if (filename3): pdfFiles.append(os.path.join(app.config['UPLOAD_FOLDER'],

```
filename3))

pdfFiles.sort(key=str.lower) print("Pdf Files : \n", pdfFiles) pdfWriter = PyPDF2.PdfWriter()
for filename in pdfFiles: pdfFileObj = open(filename, 'rb')
pdfReader = PyPDF2.PdfReader(pdfFileObj) for pageNum in range(0,
len(pdfReader.pages)): pageObj = pdfReader.pages[pageNum]
pdfWriter.add_page(pageObj) pdfOutput =
open(os.path.join(app.config['UPLOAD_FOLDER'], result_filename), 'wb')
pdfWriter.write(pdfOutput) pdfOutput.close() print("PDF Created Success")
def readPdf(filename): pdfFileObj = open(os.path.join(app.config['UPLOAD_FOLDER'],
filename), 'rb') # creating a pdf reader object pdfReader = PyPDF2.PdfReader(pdfFileObj)
# printing number of pages in pdf file print(len(pdfReader.pages))
# creating a page object pageObj = pdfReader.pages[0]
# extracting text from page data = pageObj.extract_text()
# closing the pdf file object pdfFileObj.close() return data
@app.route("/") def index(): return render_template("index.html")
@app.route("/about") def about(): return render_template("about.html")
@app.route("/newuser", methods=["POST","GET"]) def newuser(): try: msg="" print("Add
New User page") if request.method == 'POST':
fname = request.form['fname']
lname = request.form['lname']
uname = request.form['uname']
pwd = request.form['pwd']
email = request.form['email']
phnum = request.form['phnum']
address = request.form['address']
id = str(random.randint(1000, 9999))
json = {'id': id, 'FirstName': fname,
'LastName': lname,
'UserName': uname,
'Password': pwd,
'EmailId': email,
'PhoneNumber': phnum, 'Address': address}
```

```
db = firestore.client() newuser_
ref = db.collection('newuser')
id = json['id'] newuser_ref.document(id).set(json) print("New User Inserted Success")
msg = "New User Added Success" return render_template("newuser.html", msg=msg) except
Exception as e: return str(e)
@app.route("/adminlogin", methods=["POST","GET"])
def adminlogin(): msg="" if(request.method=="POST"):
uname = request.form["uname"]
pwd = request.form["pwd"]
if(uname=="admin" and pwd=="admin"):
return render_template("adminmainpage.html")
else:
msg="Invalid UserName/Password"
 return render_template("adminlogin.html", msg=msg)
@app.route("/logout") def logout():
return render_template("index.html")
 @app.route("/userlogin") def userlogin():
 return render_template("userlogin.html")
 @app.route("/stafflogin") def stafflogin():
return render_template("stafflogin.html")
@app.route("/services") def services(): return render_template("services.html")
@app.route("/gallery") def gallery(): return render_template("gallery.html")
@app.route("/adminaddstaff", methods=["POST","GET"])
def adminaddstaff():
try: print("Add New Staff page")
msg="" if request.method == 'POST':
fname = request.form['fname']
lname = request.form['lname']
uname = request.form['uname']
pwd = request.form['pwd']
email = request.form['email']
phnum = request.form['phnum']
address = request.form['address']
id = str(random.randint(1000, 9999))
```

```
json = {'id': id, 'FirstName': fname,
'LastName': lname,
'UserName': uname,
'Password': pwd,
'EmailId': email,
'PhoneNumber': phnum, 'Address': address}
db = firestore.client()
newuser_ref = db.collection('newstaff')
id = json['id'] newuser_ref.document(id).set(json)
return render_template("adminaddstaff.html", msg=msg)
except Exception as e:
return render_template("adminaddstaff.html", msg=str(e))
@app.route('/userlogincheck', methods=['POST'])
def userlogincheck():
try: msg=""
if request.method == 'POST':
uname = pwd = request.form['pwd']
db = firestore.client()
print("Uname : ", uname, " Pwd : ", pwd);
newdb_ref = db.collection('newuser')
dbdata = newdb_ref.get()
data = [] flag = False for doc in dbdata:
data = doc.to_dict() if(data['UserName']==uname and
data['Password']==pwd):
flag=True session['userid']=data['id']
break if(flag): print("Login Success")
return render_template("usermainpage.html")
else: return render_template("userlogin.html",
msg="UserName/Password is Invalid")
else: return render_template("userlogin.html", msg=msg)
except Exception as e:
return render_template("userlogin.html", msg=e)
@app.route("/adminviewstaffs")
def adminviewstaffs():
```

```
try: db = firestore.client()
newstaff_ref = db.collection('newstaff')
staffdata = newstaff_ref.get()
data = [] for doc in staffdata: data.append(doc.to_dict())
 print("Staff Data ", data)
return render_template("adminviewstaffs.html", data=data)
except Exception as e:
return str(e)
@app.route("/adminviewcontacts")
def adminviewcontacts(): try:
db = firestore.client()
newstaff_ref = db.collection('newcontact')
staffdata = newstaff_ref.get()
data = [] for doc in staffdata: data.append(doc.to_dict())
 return
render_template("adminviewcontacts.html", data=data)
except Exception as e:
 return str(e)
@app.route("/adminviewusers")
def adminviewusers():
try:
db = firestore.client()
 dbref = db.collection('newuser')
 userdata = dbref.get()
data = [] for doc in userdata: data.append(doc.to_dict())
print("Users Data ", data)
 return render_template("adminviewusers.html", data=data)
except Exception as e: return str(e)
@app.route("/adminviewreports")
def adminviewreports():
try:
db = firestore.client()
newstaff_ref = db.collection('newprescription')
 staffdata = newstaff_ref.get()
```

```
data = [] for doc in staffdata: data.append(doc.to_dict())
 return render_template("adminviewreports.html", data=data)
 except Exception as e:
 return str(e)
@app.route("/userupdateprescription",methods=['POST','GET'])
def userupdateprescription():
msg = "" if request.method == 'POST':
userid =  session['userid']
hname = request.form['hname']
dname = request.form['dname']
reason = request.form['reason']
 comments = request.form['comments']
 id = str(random.randint(1000, 9999))
 print("Insert Prescription")

file1 = request.files['file1']
file2 = request.files['file2']
file3 = request.files['file3']
filename1 = None filename2 = None filename3 = None
result_filename='Result_Filename'+ str(id) + ".pdf" if file1.
filename == '': return redirect(request.url) if file1:
 filename1 = "File1" + str(id) + ".pdf"
file1.save(os.path.join(app.config['UPLOAD_FOLDER'], filename1))
 if file2: filename2 = "File2" + str(id) + ".pdf"
file2.save(os.path.join(app.config['UPLOAD_FOLDER'], filename2))
 if file3: filename3 = "File3" + str(id) + ".pdf"
file3.save(os.path.join(app.config['UPLOAD_FOLDER'], filename3))
 json = {'id': id, 'UserId':userid, 'HospitalName': hname, 'DoctorName': dname, 'Comments':
comments,
'Reason': reason, 'File1': filename1, 'File2': filename2, 'File3': filename3,
'Result_Filename':result_filename}
combinePdfs(filename1, filename2, filename3, result_filename)
 db = firestore.client()
newuser_ref = db.collection('newprescription')
```

```
id = json['id'] newuser_ref.document(id).set(json)
msg="Prescription Uploaded Success" return
render_template("userupdateprescription.html", msg=msg)
else: return render_template("userupdateprescription.html",
msg=msg)
@app.route("/userviewprofile")
def userviewprofile():
try: userid = session['userid']
db = firestore.client()
dbref = db.collection('newuser')
userdata = dbref.get()
data = {} for doc in userdata:
temp = doc.to_dict() print(f'{doc.id} =>
{doc.to_dict()}')
if(temp['id']==userid): data = doc.to_dict()
break return render_template("userviewprofile.html", row=data)
except Exception as e:
return str(e)
@app.route("/usercheckbehaviour")
def usercheckbehaviour():
return render_template("usercheckbehaviour.html")
@app.route("/userviewreports")
def userviewreports():
return render_template("userviewreports.html")
@app.route("/userviewprescriptions")
def userviewprescriptions():
try: userid = session['userid']
db = firestore.client() newstaff_
ref = db.collection('newprescription')
staffdata = newstaff_ref.get()
data = [] for doc in staffdata: if(doc.to_dict()
['UserId']==userid): data.append(doc.to_dict())
return render_template("userviewprescriptions.html", data=data)
except Exception as e: return str(e)
```

```
@app.route("/userviewdetails",methods=['POST','GET'])
 def userviewdetails(): try: args = request.args
filename = args['filename'] data=readPdf(filename)
return render_template("userviewdetails.html", data=data)
except Exception as e: return str(e)
@app.route("/adminviewdetails",methods=['POST','GET'])
def adminviewdetails():
try: args = request.args filename = args['filename']
data=readPdf(filename)
 return render_template("adminviewdetails.html", data=data)
except Exception as e: return str(e)
@app.route("/contact",methods=["POST","GET"])
def contact(): try: print("Add New Contact page")
msg="" if request.method == 'POST':
cname = request.form['cname']
message = request.form['message']
email = request.form['email']
phnum = request.form['phnum']
id = str(random.randint(1000, 9999))
json = {'id': id,'ContactName':
cname, 'EmailId': email,
 'PhoneNumber':
 phnum, 'Message': message}
db = firestore.client()
db_ref = db.collection('newcontact')
db_ref.document(id).set(json)
msg="Contact Added Success"
return render_template("contact.html", msg=msg)
except Exception as e:
return render_template("contact.html",
 msg=str(e))
f__name__== '_main__':
 app.run(debug=True);
```

APPENDIX-B

SCREENSHOTS

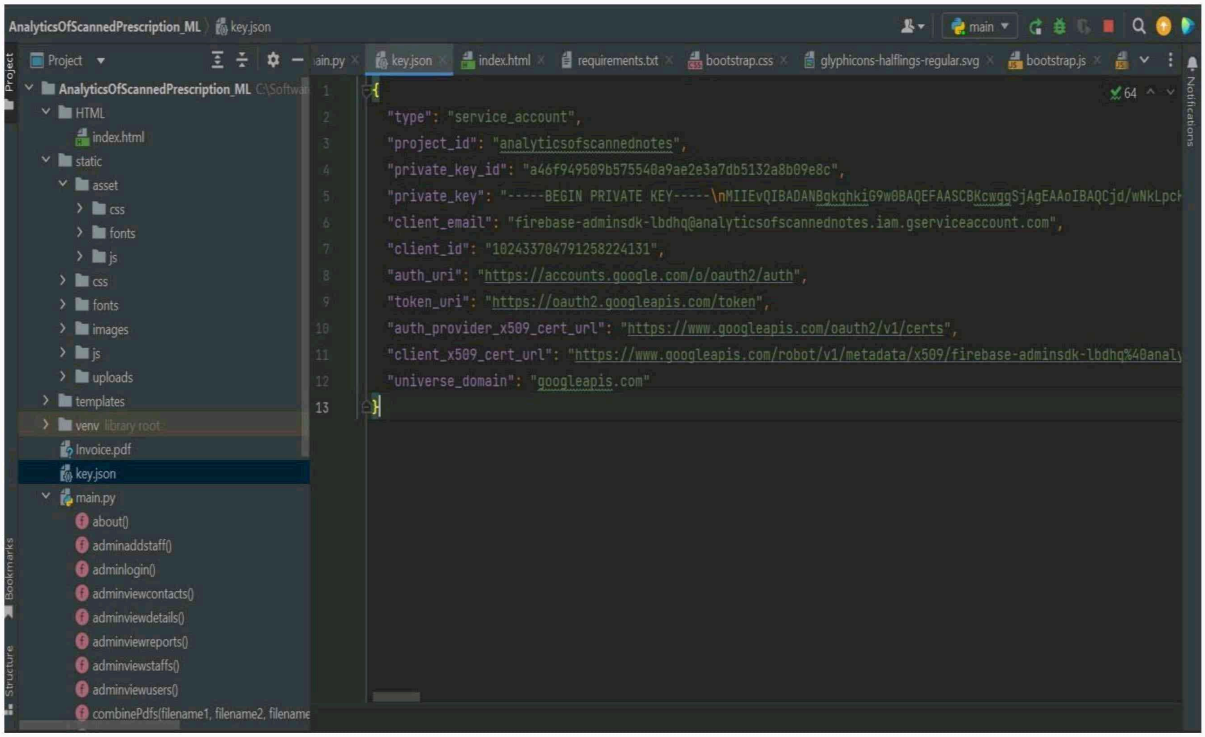


Fig Appendix- B.1 (13.1)

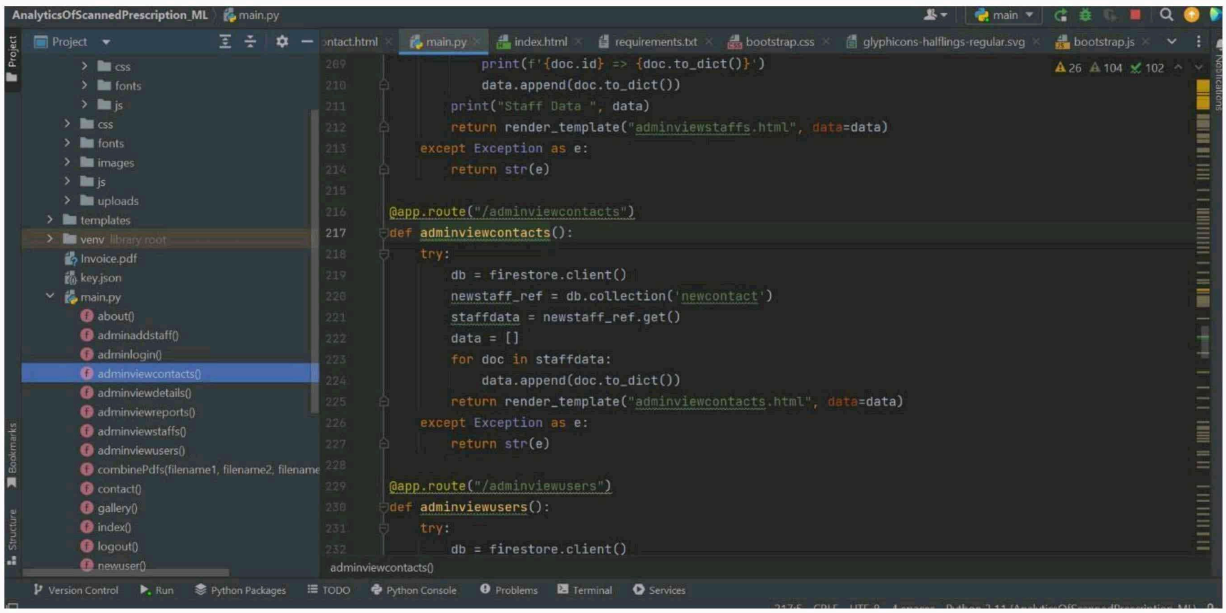


Fig Appendix- B.2(13.2)

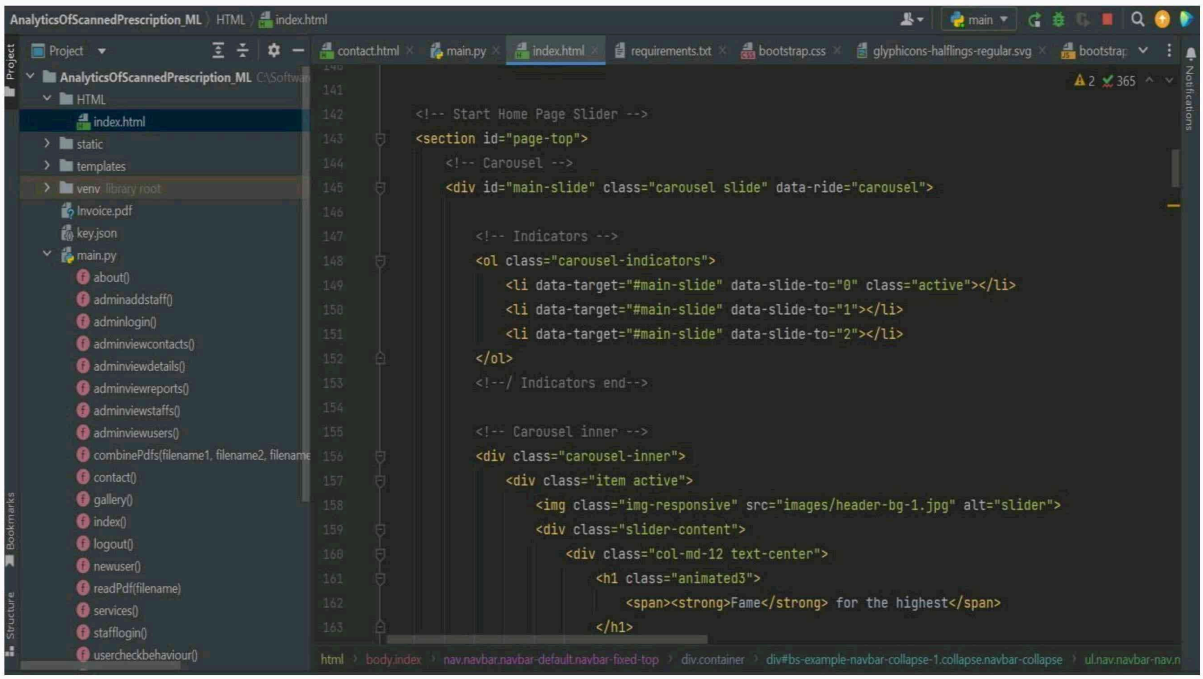


Fig Appendix- B.3 (13.3)

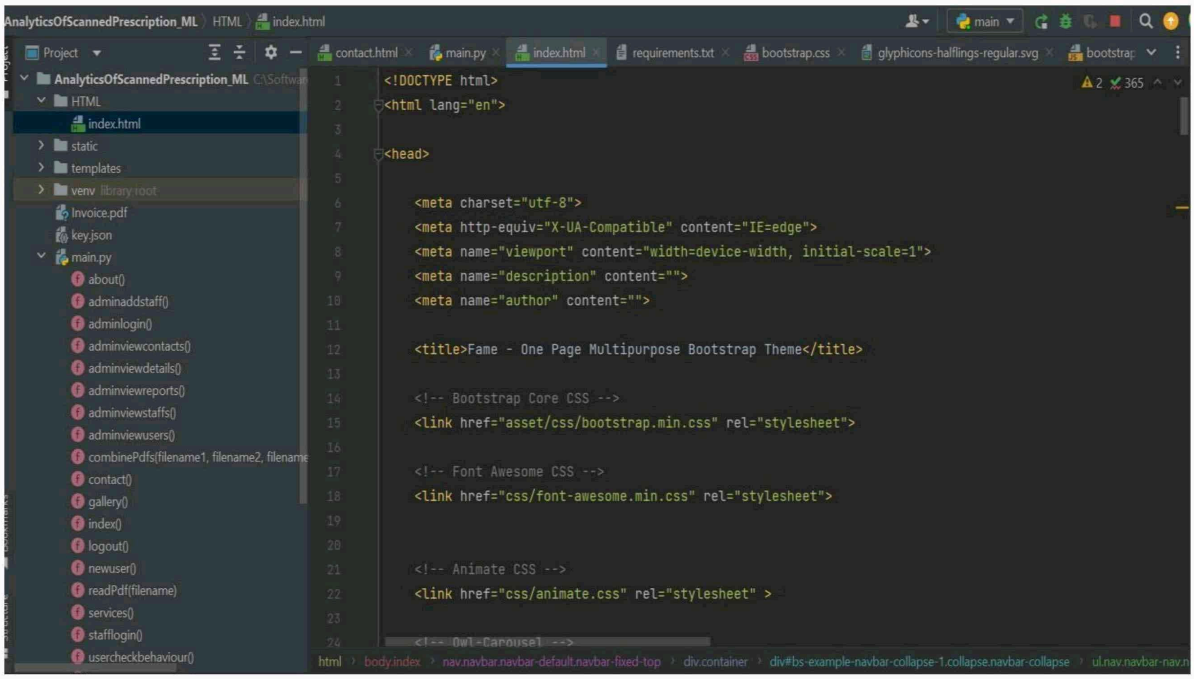
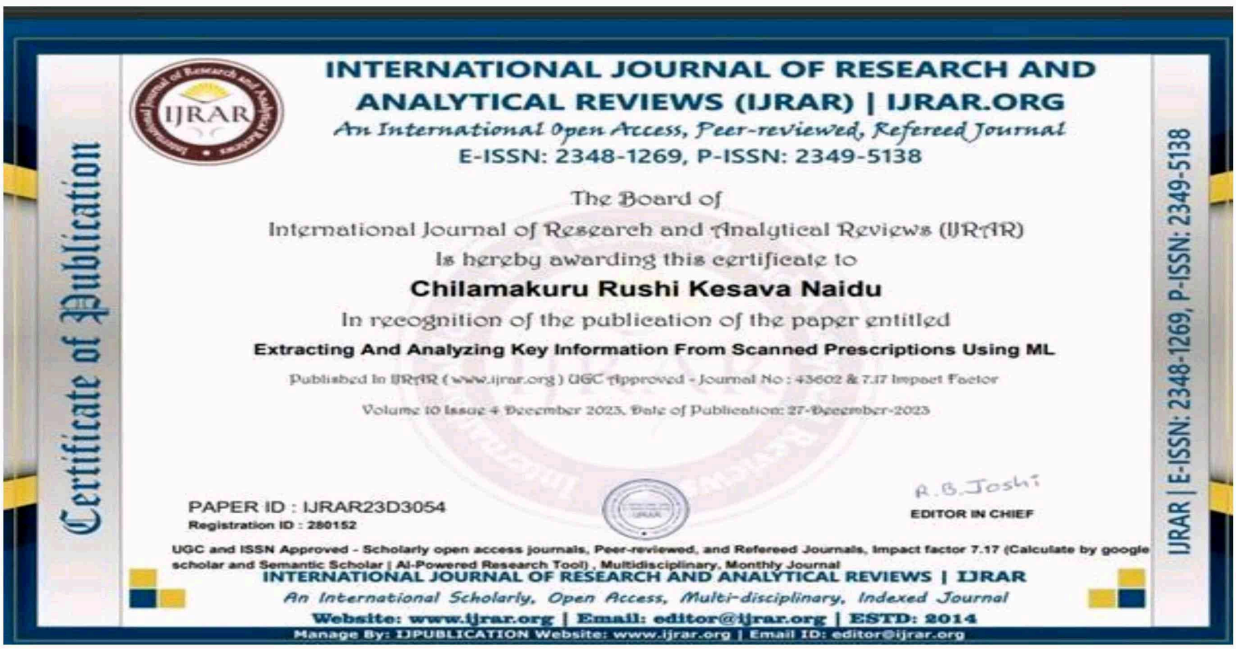
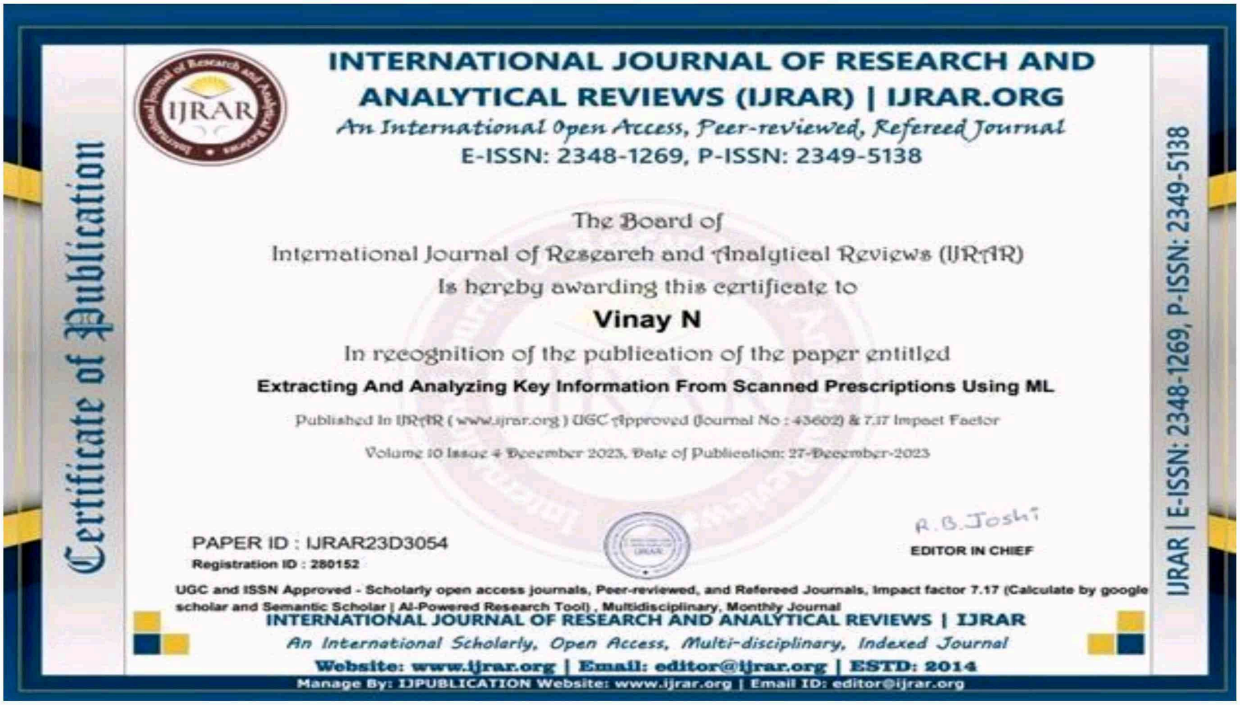


Fig Appendix- B.4 (13.4)

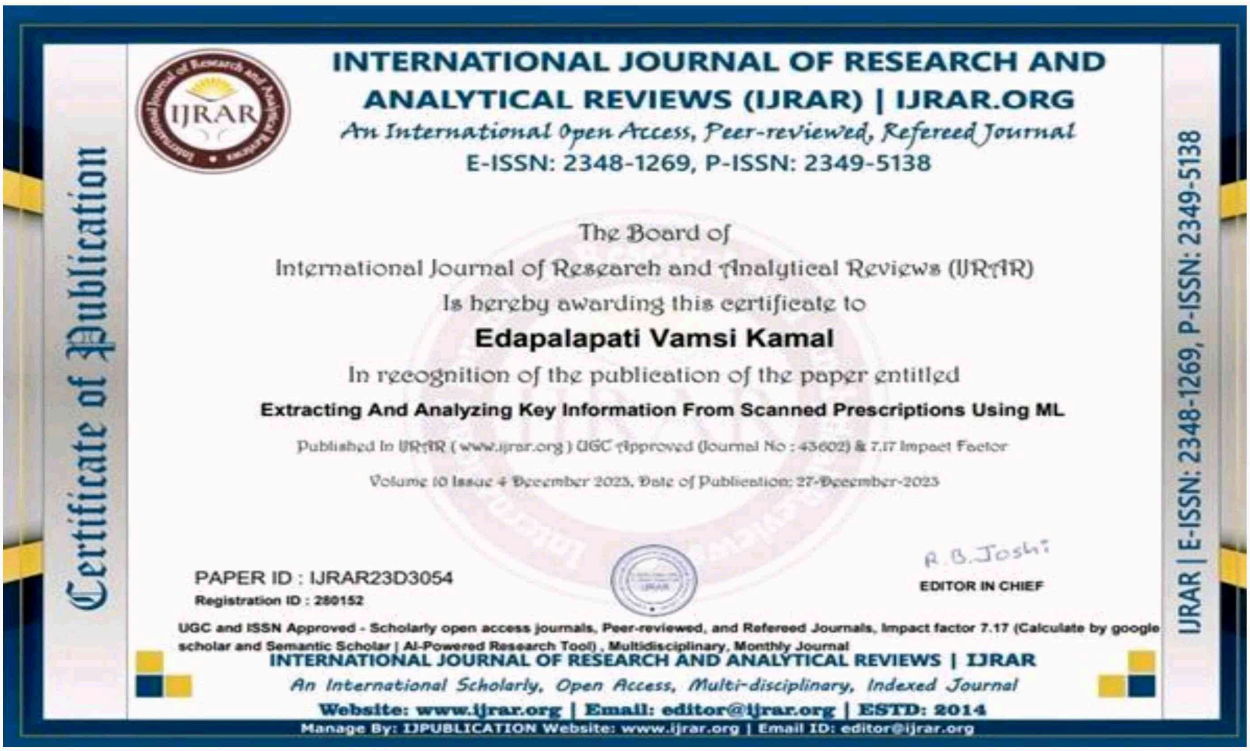
APPENDIX-C

ENCLOSURES




Certificate of Publication
INTERNATIONAL JOURNAL OF RESEARCH AND ANALYTICAL REVIEWS (IJRAR) | IJRAR.ORG
An International Open Access, Peer-reviewed, Refereed Journal
E-ISSN: 2348-1269, P-ISSN: 2349-5138
The Board of
International Journal of Research and Analytical Reviews (IJRAR)
Is hereby awarding this certificate to
Vinay N
In recognition of the publication of the paper entitled
Extracting And Analyzing Key Information From Scanned Prescriptions Using ML
Published In IJRAR (www.ijrar.org) UGC Approved (Journal No : 43602) & 7.17 Impact Factor
Volume 10 Issue 4 December 2023, Date of Publication: 27-December-2023
PAPER ID : IJRAR23D3054
Registration ID : 280152
R.B.Joshi
EDITOR IN CHIEF
UGC and ISSN Approved - Scholarly open access journals, Peer-reviewed, and Refereed Journals, Impact factor 7.17 (Calculate by google scholar and Semantic Scholar | AI-Powered Research Tool), Multidisciplinary, Monthly Journal
INTERNATIONAL JOURNAL OF RESEARCH AND ANALYTICAL REVIEWS | IJRAR
An International Scholarly, Open Access, Multi-disciplinary, Indexed Journal
Website: www.ijrar.org | Email: editor@ijrar.org | ESTD: 2014
Manage By: IJPUBLICATION Website: www.ijrar.org | Email ID: editor@ijrar.org
IJRAR | E-ISSN: 2348-1269, P-ISSN: 2349-5138


Certificate of Publication
INTERNATIONAL JOURNAL OF RESEARCH AND ANALYTICAL REVIEWS (IJRAR) | IJRAR.ORG
An International Open Access, Peer-reviewed, Refereed Journal
E-ISSN: 2348-1269, P-ISSN: 2349-5138
The Board of
International Journal of Research and Analytical Reviews (IJRAR)
Is hereby awarding this certificate to
Gundlapalli Yaswanth
In recognition of the publication of the paper entitled
Extracting And Analyzing Key Information From Scanned Prescriptions Using ML
Published In IJRAR (www.ijrar.org) UGC Approved (Journal No : 43602) & 7.17 Impact Factor
Volume 10 Issue 4 December 2023, Date of Publication: 27-December-2023
PAPER ID : IJRAR23D3054
Registration ID : 280152
R.B.Joshi
EDITOR IN CHIEF
UGC and ISSN Approved - Scholarly open access journals, Peer-reviewed, and Refereed Journals, Impact factor 7.17 (Calculate by google scholar and Semantic Scholar | AI-Powered Research Tool), Multidisciplinary, Monthly Journal
INTERNATIONAL JOURNAL OF RESEARCH AND ANALYTICAL REVIEWS | IJRAR
An International Scholarly, Open Access, Multi-disciplinary, Indexed Journal
Website: www.ijrar.org | Email: editor@ijrar.org | ESTD: 2014
Manage By: IJPUBLICATION Website: www.ijrar.org | Email ID: editor@ijrar.org
IJRAR | E-ISSN: 2348-1269, P-ISSN: 2349-5138


Certificate of Publication
INTERNATIONAL JOURNAL OF RESEARCH AND ANALYTICAL REVIEWS (IJRAR) | IJRAR.ORG
An International Open Access, Peer-reviewed, Refereed Journal
E-ISSN: 2348-1269, P-ISSN: 2349-5138
The Board of
International Journal of Research and Analytical Reviews (IJRAR)
Is hereby awarding this certificate to
Edapalapati Vamsi Kamal
In recognition of the publication of the paper entitled
Extracting And Analyzing Key Information From Scanned Prescriptions Using ML
Published In IJRAR (www.ijrar.org) UGC Approved (Journal No : 43602) & 7.17 Impact Factor
Volume 10 Issue 4 December 2023, Date of Publication: 27-December-2023
PAPER ID : IJRAR23D3054
Registration ID : 280152
R.B.Joshi
EDITOR IN CHIEF
UGC and ISSN Approved - Scholarly open access journals, Peer-reviewed, and Refereed Journals, Impact factor 7.17 (Calculate by google scholar and Semantic Scholar | AI-Powered Research Tool) , Multidisciplinary, Monthly Journal
INTERNATIONAL JOURNAL OF RESEARCH AND ANALYTICAL REVIEWS | IJRAR
An International Scholarly, Open Access, Multi-disciplinary, Indexed Journal
Website: www.ijrar.org | Email: editor@ijrar.org | ESTD: 2014
Manage By: IJPUBLICATION Website: www.ijrar.org | Email ID: editor@ijrar.org
IJRAR | E-ISSN: 2348-1269, P-ISSN: 2349-5138

and Analyzing Key Information From Scanned Prescriptions using MLA

ORIGINALITY REPORT

16%	14%	8%	10%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to M S Ramaiah University of Applied Sciences Student Paper	5%
2	pdffox.com Internet Source	4%
3	www.coursehero.com Internet Source	1%
4	pdfcoffee.com Internet Source	1%
5	hig.diva-portal.org Internet Source	1%
6	www.scribd.com Internet Source	1%
7	Submitted to University of Kurdistan Hawler Student Paper	<1%
8	Submitted to Midlands State University Student Paper	<1%
	Submitted to University of Dubai	

9	Student Paper	<1 %
10	www.programcreek.com Internet Source	<1 %
11	Submitted to Federation University Student Paper	<1 %
12	github.com Internet Source	<1 %
13	www.gatransplant.org Internet Source	<1 %
14	isownig.org Internet Source	<1 %
15	Submitted to JNTUA College of Engineering, Anantapur Student Paper	<1 %
16	Submitted to Griffth University Student Paper	<1 %
17	Submitted to University of New South Wales Student Paper	<1 %
18	noexperiencenecessarybook.com Internet Source	<1 %
19	www.slideshare.net Internet Source	<1 %

20	Marimuthu Karuppiah, T. V. Ramana, Rajanikanta Mohanty, Ganesh Gopal Devarajan, Senthil Murugan Nagarajan. "UIoTN-PMSE: Ubiquitous IoT network-based predictive modeling in smart environment", International Journal of Communication Systems, 2023 Publication	<1 %
21	openlab.ncl.ac.uk Internet Source	<1 %
22	ida3.cn Internet Source	<1 %
23	sophia.stkate.edu Internet Source	<1 %
24	stackoverflow.com Internet Source	<1 %
25	iarjset.com Internet Source	<1 %
26	Submitted to BPP College of Professional Studies Limited Student Paper	<1 %
27	gitlab.sliit.lk Internet Source	<1 %

Exclude quotes	On	Exclude matches	Off
Exclude bibliography	On		